



# ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

Título del proyecto:

DISEÑO DE UNA APLICACIÓN PARA EL ESTUDIO DE LA  
BIOCINÉTICA EN LOS ACCIDENTES DE TRÁFICO

Alumno: Iñaki Nuin Amado  
Tutor: Edurne Barrenechea Tartas  
Pamplona, Junio de 2010



# INDICE

<b>1. INTRODUCCIÓN.....</b>	<b>4</b>
1.1 – Introducción .....	5
1.2 - Estado del arte.....	6
1.3 - Objeto del Proyecto .....	7
1.4 - Planteamiento de la solución .....	7
1.5 - Fases en la realización del proyecto.....	8
<b>2. ANÁLISIS DE REQUISITOS .....</b>	<b>11</b>
2.1 – Requisitos Hardware.....	12
2.2 – Requisitos Software .....	12
2.2.1 – Requisitos de Apariencia y GUI .....	12
2.2.2 – Requisitos Funcionales.....	13
2.2.3 – Requisitos No funcionales.....	14
2.2.4 – Requisitos de almacenamiento de la información.....	15
2.3 – Objetivos del sistema .....	17
2.4 – Casos De Uso.....	18
2.5 – Diagrama Entidad-Relación.....	29
<b>3. DISEÑO .....</b>	<b>30</b>
3.1 – Diseño relacional de la base de datos .....	31
3.2 – Diseño de las interfaces .....	34
<b>4. IMPLEMENTACIÓN .....</b>	<b>42</b>
4.1 – Herramientas utilizadas (Visual Studio .NET, MySQL Server 6.0).....	43
4.2 – Incidencias y soluciones aplicadas.....	44
<b>5. INSTALACIÓN Y CONFIGURACIÓN.....</b>	<b>54</b>
5.1 – Instalación y Configuración.....	55
<b>6. CONCLUSIONES Y BIBLIOGRAFÍA .....</b>	<b>65</b>
6.1 – Conclusiones .....	66
6.2 – Bibliografía.....	67

# **Capítulo 1**

## **INTRODUCCIÓN**

## 1.1 - Introducción

Los accidentes de tráfico son un fenómeno que por mucho que se intenten disminuir, con nuevas reglamentaciones, dígase el carnet por puntos por ejemplo, o con reformas orientadas a la seguridad como el mayor control en temas tan preocupantes como el exceso de velocidad y la conducción bajo los efectos de las drogas y/o el alcohol, siguen siendo una de las mayores causas de mortalidad en la población. Sin ir más lejos el año pasado sesgaron la vida de 1.897 personas, en un número de accidentes que cuadriplica la cifra anterior.

Uno de los motivos por los que este número de muertos no se acerca al de accidentes es la pronta atención de los servicios médicos de urgencias, que en colaboración con los cuerpos de seguridad del estado son capaces de atender un aviso en un tiempo ínfimo.

Los criterios de actuación en caso de producirse un accidente de tráfico suelen comenzar por un aviso a los organismos de seguridad, los cuales son los primeros en llegar al lugar del suceso y por lo tanto son los primeros en evaluar la gravedad de los ocupantes del o los vehículos, sabiendo que sus conocimientos médicos, sin pretender entrar en generalizaciones, son bastante reducidos. Una vez evaluada la situación avisan a los servicios médicos, que según la información que se les haya transmitido elegirán un tipo de equipamiento médico, como por ejemplo hablaremos de una ambulancia completamente equipada como UCI móvil o una más sencilla con lo necesario para realizar atenciones que no incluyan riesgo elevado.

La valoración de los accidentados por parte de personal no titulado en medicina, es un tema que trae mucha polémica, ya que las consecuencias de un error en la apreciación del estado de cada uno de los ocupantes de los vehículos implicados pueden desembocar en causas trágicas.

En este punto incluiremos la tesis del doctor Diego Reyero que tras un estudio exhaustivo del material facilitado por las distintas policías y servicios médicos, ha propuesto un método de valoración de la gravedad de los accidentados basado en un conjunto de factores apreciables a simple vista, como lo son las deformaciones del vehículo en el que circulaban, la posición que ocupaba cada uno, si llevaban el cinturón de seguridad, etc. A este método lo llamaremos IDE (Índice de Deformación Estructural).

Como he expuesto antes Diego Reyero tuvo que trabajar, en la actualidad sigue trabajando, con mucha información obtenida tanto

por organizaciones médicas como de seguridad del estado, lo cual dificultaba mucho su estudio y el desglose en los parámetros necesarios para desarrollar la metodología que defiende en su tesis. Por ello, me propusieron la realización de una aplicación que almacenase de forma sencilla, consistente y ordenada todos los datos que Diego fuese tratando. Facilitando de esta forma su archivado y consulta cuando fuera necesario.

Aquí comienza el desarrollo de la aplicación cuyo título es "DISEÑO DE UNA APLICACIÓN PARA EL ESTUDIO DE LA BIOCINÉTICA EN LOS ACCIDENTES DE TRÁFICO".

## 1.2 - Estado del arte

Actualmente existen muchas herramientas para el almacenamiento y tratamiento de datos, como por ejemplo lo son el Excel o el paquete estadístico spss. Ambas son aplicaciones registradas, cuya forma visual he intuitiva de introducción de los datos las hacen perfectas para usuarios inexpertos. Permitiendo una relativa organización de toda la información que necesitan guardar para su posterior evaluación.

Los problemas que plantea la utilización de las dos herramientas anteriormente mencionadas son principalmente dos:

- 1- Gran redundancia en los datos: si un coche tiene cuatro ocupantes repetiremos cuatro veces los datos correspondientes al vehículo, lo que provoca un mayor consumo de memoria por parte del equipo en el que trabajemos.
- 2- Ralentiza el trabajo de tratamiento y estudio de la información: el almacenamiento innecesario de datos dificulta sobremanera la búsqueda de características y accidentes concretos.

A parte de estas aplicaciones genéricas el usuario puede utilizar aplicaciones más específicas y mejor adaptadas al entorno del almacenamiento y la consulta, introduciendo en pequeña escala modelos de desarrollo de bases de datos, un ejemplo en concreto sería el MsAccess. El inconveniente es la dificultad para usuarios inexpertos obligándoles a un pequeño estudio de dichas aplicaciones antes de su uso, lo que para personas no iniciadas en el mundillo informático, puede resultar una gran dificultad, muchas veces muy difícil de superar.

## 1.3 - Objeto del Proyecto

Con el objetivo de superar las dificultades y solucionar los problemas planteados en el apartado anterior planteamos el desarrollo de un nuevo proyecto que ofrezca una aplicación para el almacenamiento y consulta específico para accidentes de tráfico y sus ocupantes, con un control de datos y una ordenación supervisada y evaluada por el médico Diego Reyero.

La solución está basada en una base de datos que cumpla el modelo relacional y una interfaz gráfica, sencilla y robusta, perfecta para usuarios con conocimientos informáticos básicos.

## 1.4 - Planteamiento de la solución

En primer lugar, después de tener una extensa entrevista con Diego en la que me planteo sus necesidades y de observar el sistema informático en el que se maneja habitualmente, comencé a plantearme las funcionalidades básicas que deberían completar el proyecto. He diseñado dos aplicaciones independientes que operan sobre la misma base de datos, siendo una el programa principal y la otra un administrador. Estas dos aplicaciones van a tener las siguientes características generales:

1. Ofrecer, por parte del programa principal, las funcionalidades de almacenamiento y consulta tanto de accidentes como de usuarios y colisiones, dividiéndolas en los siguientes formularios: Alta accidente, Consultar/Actualizar datos accidente y Consultar choque.
2. Administrar la base de datos de forma sencilla y eficaz, consiguiendo que con un solo clic podamos crear, eliminar, poblar o volcar los datos almacenados en ficheros de texto.
3. Estructurar la arquitectura del proyecto de tal manera que sea posible la actuación de dos aplicaciones diferentes sobre la misma base de datos sin provocar errores de ejecución.
4. Mantener la organización evitando redundancias de datos, tablas sin relación y el almacenamiento innecesario de campos irrelevantes para el estudio de la tesis.

5. Cumplir patrones estructurales en la programación y seguimiento del modelo relacional en el desarrollo de la base de datos.

## 1.5 - Fases en la realización del proyecto

Para la realización del proyecto descrito anteriormente se han seguido una serie de pasos que pueden resumirse en las siguientes fases:

Inicialmente, realice una extensa entrevista con el cliente al que van destinadas las aplicaciones que desarrollo en mi proyecto, con el objetivo de tener conocimiento de su metodología de trabajo y comprender todos los tecnicismos médicos que debo incluir. Todo esto se ha de tener en cuenta a la hora de realizar el análisis funcional de la aplicación.

A continuación, se ha definido el proyecto, de la misma manera que lo hemos definido en el apartado anterior, planteando una solución detallada sobre qué aspectos va a recoger en la aplicación y todas las partes y funcionalidades que debe cumplir.

Una vez definido con detalle el proyecto, se ha procedido a la fase de Análisis de Requisitos de Hardware y Software de nuestra aplicación; es decir, por un lado se han analizado necesidades hardware, como el tipo de máquina que vamos a necesitar en nuestro desarrollo y en el entorno en el que se debe desarrollar y probar. Por otro lado, se han analizado detalladamente todos los requisitos software que la aplicación debe cumplir, funcionales y no funcionales, para que tenga las características finales deseadas.

Tras definir los requisitos, se ha realizado el Análisis orientado a objetos, en el que se ha especificado la estructura de todo el sistema, describiendo cuáles son las opciones que va a poder ver el usuario que trabaje con ella y en la que nos hemos basado para las posteriores fases de Diseño e Implementación.

A continuación, y a partir del análisis anterior, se ha procedido a realizar la fase de Diseño. En esta fase se ha diseñado por un lado la Base de Datos relacional con las tablas correspondientes a la misma y sus relaciones, y por otro lado las interfaces de las diferentes aplicaciones de las que se compone el proyecto. Para el diseño de la interfaz de la aplicación de almacenamiento y consulta de accidentes de tráfico se ha intentado en todo momento conseguir un entorno amigable, sencillo de utilizar. De manera similar se ha

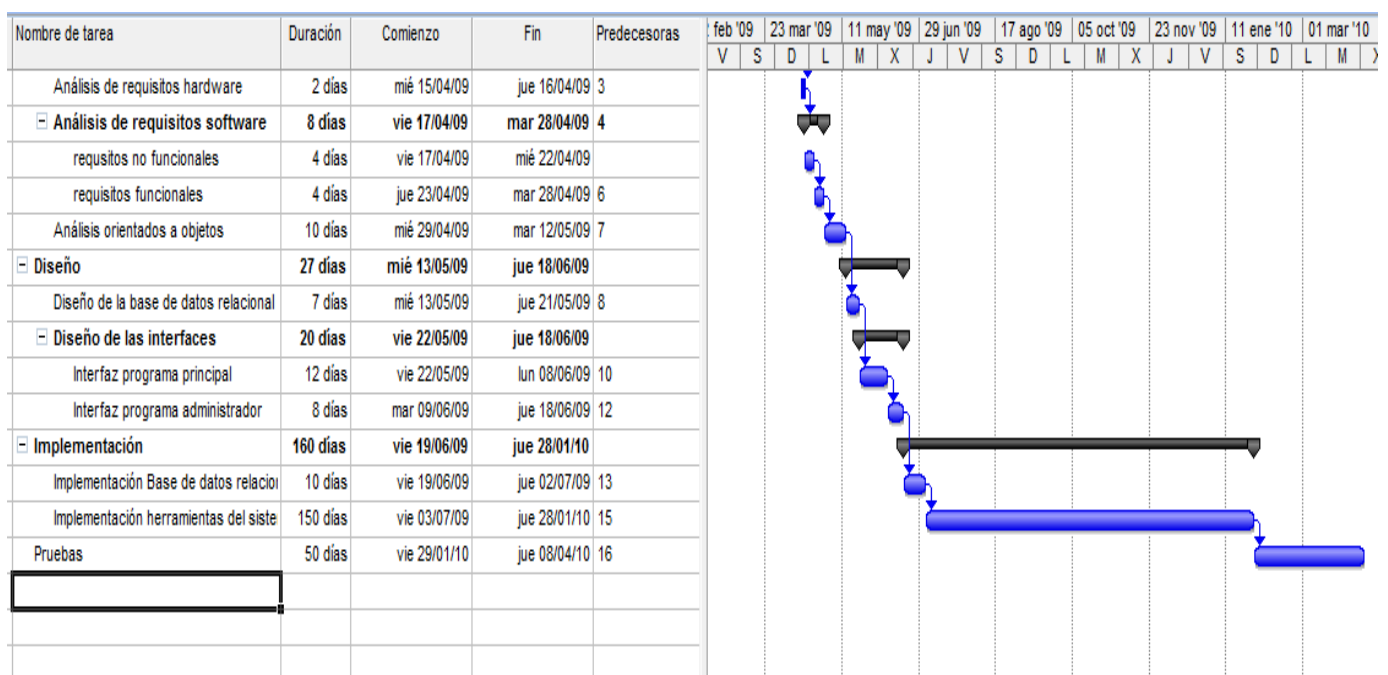


intentado que la aplicación para la gestión y administración de la base de datos, tenga una interfaz amigable y sencillo de utilizar, que además muestre en todo momento las opciones que tiene el usuario, y que la forma de trabajar con ella sea de manera clara y visible. Proveyendo de la seguridad y robustez necesaria en el tratamiento de los datos.

Tras finalizar el análisis y el diseño, se ha realizado la fase de Implementación de todas las aplicaciones que componen el sistema. Esta implementación ha consistido en, por un lado, construir la base de datos a partir del diseño relacional de la misma, y por otro lado, el desarrollo de todas las herramientas que componen el sistema, eligiendo el lenguaje de programación y la metodología más adecuada, según las necesidades de la misma.

Una vez desarrollado todo el sistema, y para dar finalizado el proyecto, se ha llevado a cabo una fase de pruebas, un proceso de integración y pruebas sobre el entorno necesario para la puesta en marcha de la herramienta.

Para poder apreciar de forma más gráfica el proceso de desarrollo del proyecto he preparado una estimación mediante la herramienta Microsoft Project. Los tiempos son aproximados y las relaciones entre las tareas son de fin-comienzo ya que yo era el único desarrollador (cuando terminaba una tarea comenzaba con la siguiente):



Si deseas acceder al documento original con la estimación del proyecto pulsa Control (CTRL) y haz clic sobre la imagen.

***\*Nota: Debes tener instalado el programa Microsoft Project.***

## **Capítulo 2**

# **ANÁLISIS DE REQUISITOS**

## 2.1 - Requisitos Hardware

Los requisitos citados a continuación vienen impuestos principalmente por el entorno donde se desarrolla la aplicación.

R.H.1: Deseamos que la aplicación sea libre y gratuita, es decir, que los usuarios finales no tengan que realizar ningún desembolso a la hora de trabajar con este software.

Como editor hemos decidido utilizar Visual Studio .net, que aun siendo una aplicación de pago, únicamente es utilizada por el desarrollador. Los usuarios finales utilizan el ejecutable facilitado por este.

Para la gestión de bases de datos seleccione MySQL, software que si deberán instalar los usuarios, pero que debido a su carácter GNU es gratuito (excepto para aquellas empresas que decidan realizar un uso privativo de MySQL, que si deben comprar una licencia). Lo cual permite cumplir el objetivo de gratuidad del proyecto.

R.H.2: Se pretende que la aplicación pueda ser utilizada por la mayoría de los usuarios, por lo que se utilizará un sistema operativo Windows, ya que es más accesible, más sencillo, y su uso está más extendido.

## 2.2 - Requisitos Software

Los requisitos software son aquellos que la aplicación debe cumplir, requisitos funcionales y no funcionales que se requieren para que la aplicación tenga las características finales que cualquier usuario final desea.

### 2.2.1 - Requisitos de Apariencia y GUI

Tanto la aplicación principal como la aplicación de administración de la base de datos y su contenido, van a ser intuitivas e interactivas, además de sencillas de utilizar, por lo que el usuario no va a necesitar de ningún tipo de conocimiento adicional para poder utilizarlas. El aspecto de las aplicaciones será clave para conseguir este objetivo.

R.A.1: Las opciones disponibles para el usuario quedarán claras en todo momento, aquellas que en un instante dado no sea conveniente que se seleccionen o modifiquen se desactivarán para

que no puedan ser activadas ni desactivadas, o en su defecto se informará al usuario de que la opción seleccionada no es correcta.

R.A.2: Si el usuario introduce datos de una accidente o de un usuario que ya han sido almacenados en la base de datos se le informará mediante un mensaje.

R.A.3: El usuario, en el momento en que quiere visualizar cualquier accidente o colisión, así como los datos de los implicados en el accidente, se le facilitarán todas las posibilidades almacenadas en la base de datos con respecto a los campos que debe rellenar obligatoriamente para realizar la consulta.

R.A.4: Si a la hora de realizar cualquier tipo de consulta con unos datos que no devuelven ningún resultado, se indicará al usuario que no hay resultados almacenado en la base de datos.

R.A.5: Si el usuario administrador decide realizar operaciones que hagan peligrar la integridad tanto de la base de datos como de su contenido, se desplegarán ventanas informativas que le advertirán de los resultados de su acción, permitiéndole cancelar la acción o continuar con ella.

## 2.2.2 - Requisitos Funcionales

Estos requisitos hacen referencia a las características que la herramienta debe satisfacer respecto al modo de trabajo.

### **La nueva aplicación principal:**

1. Ofrecer una forma sencilla e intuitiva de inserción de los datos, indicando claramente los campos obligatorios a rellenar.
2. Simplificar la arquitectura y reducir posibles puntos de fallo.
3. Facilitar las consultas sobre la base de datos aportando las posibilidades almacenadas con respecto a los campos obligatorios a rellenar para realizarlas.
4. Permitir la inserción y consulta de colisiones, tanto simples como múltiples, incluyendo todos los ocupantes de los vehículos, a través de una interfaz intuitiva y sencilla.

### **La aplicación de administración de la base de datos y su contenido:**

1. Permitir eliminar y crear la base de datos sobre la que interactúa la aplicación de forma sencilla e intuitiva.

2. Ofrecer la posibilidad de volcar los datos almacenados en ficheros de texto, otorgando una forma sencilla y con un bajo coste de almacenamiento para realizar copias de seguridad.
3. Ofertar el poblado de la base de datos a partir de los ficheros de texto generados, con un formato concreto, en el punto 2.

### 2.2.3 - Requisitos No Funcionales

Estos requisitos son características que la herramienta debe satisfacer pero no son parte de la razón fundamental del producto. Son un conjunto de características de calidad que debemos tenerlas en cuenta a la hora de diseñar e implementar el software que vamos a desarrollar.

Para este proyecto, los requisitos no funcionales que se deben cumplir se indican a continuación.

<b>RNF-01</b>	<b>Almacenamiento</b>
<b>Objetivos asociados</b>	
<b>Requisitos asociados</b>	
<b>Descripción</b>	La base de datos de la aplicación debe estar en alta disponibilidad.
<b>Comentarios</b>	

<b>RNF-02</b>	<b>Conectividad con bases de datos</b>
<b>Objetivos asociados</b>	
<b>Requisitos asociados</b>	
<b>Descripción</b>	La aplicación debe ser robusta frente a pérdidas temporales de conexión con la base de datos.
<b>Comentarios</b>	

<b>RNF-03</b>	<b>Control de la base de datos y su contenido</b>
<b>Objetivos asociados</b>	
<b>Requisitos asociados</b>	

<b>Descripción</b>	La aplicación debe poder administrar tanto la base de datos como su contenido.
<b>Comentarios</b>	

## 2.2.4 - Requisitos de almacenamiento de la información

<b>RI-01</b>	<b>Información de los vehículos accidentados</b>
<b>Descripción</b>	El sistema deberá almacenar toda la información relacionada con los vehículos accidentados.
<b>Objetivos asociados</b>	OBJ-01 Gestionar los accidentes OBJ-03 Gestionar las consultas sobre la base de datos
<b>Datos específicos</b>	Matricula Fecha Hora NumOcupantes TipoColision VelocidadExcesiva TipoVehiculo VelocidadEstimada Airbag EuronCAP AnioMatriculacion
<b>Intervalo temporal</b>	Presente
<b>Estabilidad</b>	Crítica
<b>Comentarios</b>	

<b>RI-02</b>	<b>Información de los ocupantes de los vehículos accidentados</b>
<b>Descripción</b>	El sistema deberá almacenar toda la información relacionada con los ocupantes de los vehículos accidentados.
<b>Objetivos asociados</b>	OBJ-01 Gestionar los accidentes OBJ-03 Gestionar las consultas sobre la base de datos
<b>Datos</b>	Matricula

<b>específicos</b>	Fecha Hora Id Edad IDE GlasgowSamu GlasgowUrg GlasgowUci RTSSamu RTSUrg RTSUci ISSUrg ISSUci LesionPrincipal TipoAtencion Tce Tfacial Traquis Ttorax Tabd Tee Tbland
<b>Intervalo temporal</b>	Presente
<b>Estabilidad</b>	Crítica
<b>Comentarios</b>	

<b>RI-03</b>	<b>Información de las colisiones entre vehículos</b>
<b>Descripción</b>	El sistema deberá almacenar toda la información relacionada con las colisiones o choques entre vehículos.
<b>Objetivos asociados</b>	OBJ-02 Gestionar las colisiones entres vehículos OBJ-03 Gestionar las consultas sobre la base de datos
<b>Datos específicos</b>	Matricula Fecha Hora Matricula2
<b>Intervalo temporal</b>	Presente
<b>Estabilidad</b>	Crítica
<b>Comentarios</b>	



## 2.3 - Objetivos del sistema

<b>OBJ-01</b>	<b>Gestionar los accidentes</b>
<b>Descripción</b>	El sistema permite realizar la gestión de toda la información relacionada con los vehículos y sus ocupantes de forma segura y ordenada.
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	

<b>OBJ-02</b>	<b>Gestionar las colisiones entre vehículos</b>
<b>Descripción</b>	El sistema permite realizar la gestión de todas las colisiones entre vehículos, mostrando el tipo de colisión y los datos de los ocupantes de los vehículos.
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	

<b>OBJ-03</b>	<b>Gestionar las consultas sobre la base de datos</b>
<b>Descripción</b>	El sistema oferta todas las posibles consultas, tanto en referencia a los vehículos y sus ocupantes, como a las colisiones entre estos.
<b>Estabilidad</b>	Crítica
<b>Comentarios</b>	

<b>OBJ-04</b>	<b>Administración de la base de datos</b>
<b>Descripción</b>	El sistema permite al usuario administrador eliminar y crear la base de datos.
<b>Estabilidad</b>	Crítica
<b>Comentarios</b>	La eliminación implica la pérdida de los datos almacenados, por ello introduzco el volcado sobre ficheros de texto.

<b>OBJ-05</b>	<b>Control del contenido de la base de datos</b>
<b>Descripción</b>	El sistema permite volcar sobre ficheros de texto los datos de la base de datos y el posterior poblado de esta usando los ficheros generados.

<b>Estabilidad</b>	Crítica
<b>Comentarios</b>	Posibilito la creación de copias de seguridad de los datos de forma sencilla y con un consumo de recursos de almacenamiento mínimo.

## 2.4 - Diagramas de Casos de Uso

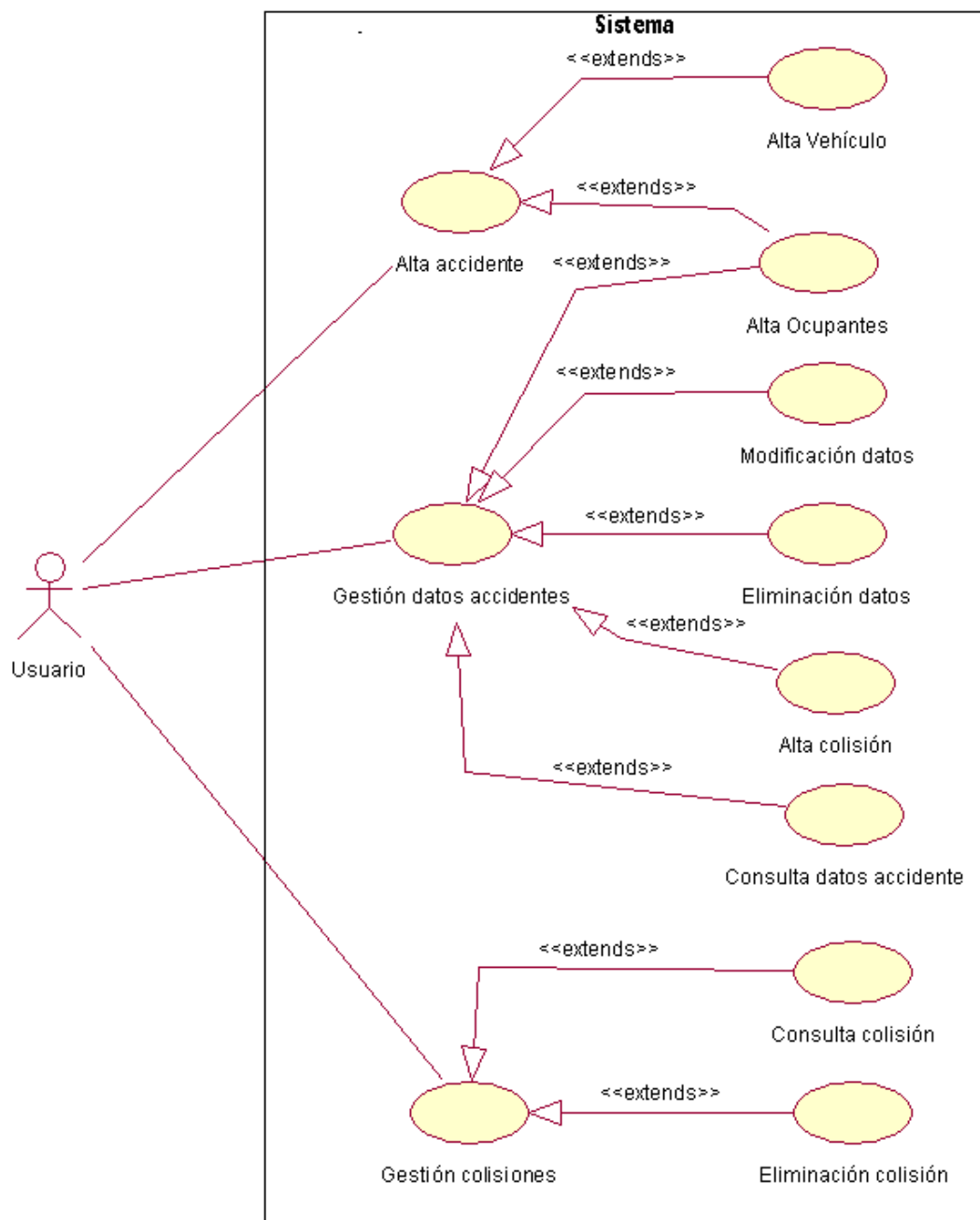


Figura 1. Diagrama de casos de uso del programa principal de gestión de accidentes

## Definición de actores

<b>ACT-01</b>	<b>Usuario</b>
<b>Descripción</b>	Representa a los usuarios sin privilegios que utilizan el programa.
<b>Comentarios</b>	Total acceso a todas las funciones del programa principal. No posee privilegios para utilizar el administrador.

## Casos de uso del sistema

Tras analizar los requisitos que la herramienta debe satisfacer se procede a analizar su funcionalidad mediante los casos de uso que vamos a exponer a continuación. Con estos diagramas conseguiremos representar la funcionalidad de cada uno de sus componentes y sus relaciones.

<b>RF-01</b>	<b>Alta Vehículo</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los accidentes	
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados	
<b>Descripción</b>	Cuando el usuario desee agregar un nuevo vehículo accidentado el proceso será el siguiente.	
<b>Precondición</b>	Los valores de matricula, fecha y hora no pueden ser iguales al de otro ya creado. Además estos y NumOcupantes no pueden ser cadena vacía.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario selecciona la opción de "Añadir Vehículo".
	<b>2</b>	Introduce los datos requeridos por el sistema.
	<b>3</b>	El sistema comprueba que los datos matricula, fecha y hora no estén duplicados, y que estos y NumOcupantes no sean vacíos. Si es así, lo almacena.
<b>Poscondición</b>	Se ha agregado un nuevo vehículo accidentado.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si por el contrario ya existe un vehículo con los valores de los tres datos anteriores, se informa del error y no se almacena.
	<b>4</b>	Si alguno de los cuatro campos indicados

		en la precondition es vacío, se informa de ello y se espera a que sean rellenados.
<b>Comentarios</b>		

<b>RF-02</b>	<b>Alta Ocupante</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los accidentes	
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados RI-02 Información de los ocupantes de los vehículos accidentados	
<b>Descripción</b>	Cuando el usuario desee agregar un nuevo ocupante a un vehículo, previamente almacenado, el proceso será el siguiente.	
<b>Precondición</b>	Los valores de matricula, fecha, hora e id no pueden ser iguales al de otro ya creado. Además estos y Edad, IDE y Tipoatención no pueden ser cadena vacía.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario selecciona la opción de "Añadir Ocupante".
	<b>2</b>	Introduce los datos requeridos por el sistema.
	<b>3</b>	El sistema comprueba que los datos matricula, fecha, hora e Id no estén duplicados, y que estos y Edad, IDE y Tipoatención no sean vacíos. Si es así, lo almacena.
<b>Poscondición</b>	Se ha agregado un nuevo ocupante.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si por el contrario ya existe un ocupante con los valores de los cuatro datos anteriores, se informa del error y no se almacena.
	<b>4</b>	Si alguno de los siete campos indicados en la precondition es vacío, se informa de ello y se espera a que sean rellenados.
<b>Comentarios</b>	Los datos matricula, fecha y hora identifican el vehículo en el que circulaba el ocupante. El valor que identifica al ocupante es Id.	

<b>RF-03</b>	<b>Modificación datos</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los accidentes OBJ-03 Gestionar las consultas sobre la base de datos.	
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados RI-02 Información de los ocupantes de los vehículos accidentados	
<b>Descripción</b>	Cuando el usuario desee modificar la información de un vehículo u ocupante ya existente el procedimiento a seguir será el siguiente	
<b>Precondición</b>	Debe haber algún vehículo con sus respectivos ocupantes en el sistema para poder modificar. Los valores matricula, fecha, hora e Id no pueden ser modificados.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario accede al apartado "Consultar/actualizar datos accidente"
	<b>2</b>	Realizar una consulta con los datos referentes al vehículo a modificar.
	<b>3</b>	Modificar cualquiera de los datos mostrados en pantalla mediante las listas desplegables o directamente escribiendo desde el teclado.
	<b>4</b>	El sistema comprueba que los nuevos datos son correctos y los almacena
<b>Poscondición</b>	Las modificaciones se almacenan con éxito.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si alguna de las modificaciones no es correcta, se notifica al usuario con un pantallazo y se devuelve a la variable modificada su valor anterior.
	<b>4</b>	Si la consulta realizada en el paso 2 no devuelve resultados, dicho vehículo ya no existe.
<b>Comentarios</b>	Las restricciones en cuanto a modificaciones son muy estrictas y de muchos tipos.	

<b>RF-04</b>	<b>Eliminación datos</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los accidentes OBJ-03 Gestionar las consultas sobre la base de datos.	
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados. RI-02 Información de los ocupantes de los vehículos accidentados.	

<b>Descripción</b>	Cuando el usuario desee eliminar un vehículo y sus respectivos ocupantes, o únicamente alguno de los ocupantes de un vehículo concreto.	
<b>Precondición</b>	El vehículo o el ocupante a eliminar deben estar almacenados.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario accede al apartado "Consultar/actualizar datos accidente"
	<b>2</b>	Realizar una consulta con los datos referentes al vehículo a eliminar.
	<b>3</b>	Pulsar el botón derecho sobre la fila a eliminar. Esta se seleccionará entera y aparecerá un menú con la opción "Eliminar fila".
	<b>4</b>	Seleccionar dicha opción y esa fila desaparece de la pantalla y de la base de datos.
<b>Poscondición</b>	Si la fila eliminada era un vehículo se eliminará de la base de datos dicho vehículo y sus ocupantes. Si era un ocupante únicamente este será borrado de la base de datos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si la consulta realizada en el paso 2 no devuelve resultados, dicho vehículo ya no existe.
<b>Comentarios</b>		

<b>RF-05</b>	<b>Alta Colisión</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestionar las colisiones entres vehículos. OBJ-03 Gestionar las consultas sobre la base de datos.	
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados. RI-02 Información de los ocupantes de los vehículos accidentados. RI-03 Información de las colisiones entre vehículos.	
<b>Descripción</b>	Cuando el usuario desee agregar una nueva colisión entre dos vehículos, previamente almacenados, el proceso será el siguiente	
<b>Precondición</b>	Los valores de fecha y hora de ambos vehículos deben ser la misma, y el atributo TipoColisión no puede ser del tipo "Vuelco sin colisión" en ninguno de los dos.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario accede al apartado "Consultar/actualizar datos accidente".

	<b>2</b>	Realizar una consulta con la fecha y la hora en la que quiere almacenar la colisión.
	<b>3</b>	El sistema devolverá los vehículos accidentados en esa fecha y esa hora. Si aparecen en pantalla dos o más vehículos se mostrara la opción de "Añadir colisión".
	<b>4</b>	El usuario seleccionara dos vehículos.
	<b>5</b>	El sistema comprueba que se cumpla todo lo explicado en la precondition, si es así, almacena la colisión.
<b>Poscondición</b>	Se ha agregado un nuevo ocupante.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si por el contrario la fecha o la hora de alguno de los dos vehículos es diferente, la colisión no es posible. El sistema se lo indicará al usuario mediante un pantallazo y no almacenara la colisión.
	<b>4</b>	Si el atributo TipoColisión es del tipo "Vuelco sin colisión" en alguno de los dos vehículos, ese accidente no fue una colisión. El sistema se lo indicara al usuario mediante un pantallazo y no almacenará la colisión.
	<b>5</b>	Si el usuario selecciona más de dos vehículos, el sistema se lo indicará al usuario mediante un pantallazo y no almacenará la colisión. Ya que las colisiones son siempre uno a uno.
<b>Comentarios</b>	También podemos mostrar colisiones múltiples, pero para almacenarlas deberemos hacerla de uno en uno.	

<b>RF-06</b>	<b>Consulta datos accidente</b>
<b>Objetivos asociados</b>	OBJ-03 Gestionar las consultas sobre la base de datos
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados RI-02 Información de los ocupantes de los vehículos accidentados
<b>Descripción</b>	Cuando el usuario desee realizar una consulta sobre la base de datos para observar los datos almacenados referentes a un accidente.
<b>Precondición</b>	Alguno de los tres campos de búsqueda debe estar relleno.

<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario accede al apartado "Consultar/actualizar datos accidente".
	<b>2</b>	Rellena alguno de los tres campos (matricula, fecha y hora). El número de campos rellena puede variar de 1 a 3.
	<b>3</b>	El sistema comprueba la base de datos con los campos rellenos en el paso 2 y muestra los resultados en pantalla.
<b>Poscondición</b>	Se ha agregado un nuevo ocupante.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si la consulta no devuelve ningún resultado, se informa al usuario mediante un pantallazo.
	<b>4</b>	Si alguno de los campos de búsqueda relleno no es correcto, se informa al usuario de cómo debe relleno dicho campo y se cancela la búsqueda.
<b>Comentarios</b>	La consulta muestra los datos del o los vehículos resultantes, para ver la información de los usuarios debemos clicar sobre alguno de los campos del vehículo.	

<b>RF-07</b>	<b>Consulta colisión</b>	
<b>Objetivos asociados</b>	OBJ-03 Gestionar las consultas sobre la base de datos.	
<b>Requisitos asociados</b>	RI-01 Información de los vehículos accidentados. RI-02 Información de los ocupantes de los vehículos accidentados. RI-03 Información de las colisiones entre vehículos.	
<b>Descripción</b>	Cuando el usuario desee realizar una consulta sobre la base de datos para observar los datos referentes a una colisión.	
<b>Precondición</b>	Los tres campos de búsqueda deben estar rellenos, ya que buscamos una colisión en concreto.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario accede al apartado "Consultar choque".
	<b>2</b>	Rellena los tres campos (matricula, fecha y hora).
	<b>3</b>	El sistema comprueba la base de datos con los campos rellenos en el paso 2 y muestra los resultados en



			pantalla.
<b>Poscondición</b>	Se ha agregado un nuevo ocupante.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	<b>3</b>	Si la consulta no devuelve ningún resultado, se informa al usuario mediante un pantallazo.	
	<b>4</b>	Si alguno de los campos de búsqueda rellenado no es correcto, se informa al usuario de cómo debe rellenarse dicho campo y se cancela la búsqueda.	
<b>Comentarios</b>	La consulta muestra los datos del vehículo resultante, así como del o los vehículos que colisionaron con él en ese accidente. Para ver la información de los usuarios debemos clickar sobre alguno de los campos del vehículo.		

RF-08	Eliminación colisión		
Objetivos asociados	OBJ-01 Gestionar las colisiones entres vehículos OBJ-03 Gestionar las consultas sobre la base de datos.		
Requisitos asociados	RI-01 Información de los vehículos accidentados. RI-02 Información de los ocupantes de los vehículos accidentados. RI-03 Información de las colisiones entre vehículos		
Descripción	Cuando el usuario desee eliminar una colisión entre vehículos.		
Precondición	La colisión a eliminar debe estar almacenada.		
Secuencia normal	Paso	Acción	
	1	El usuario accede al apartado "Consultar choque".	
	2	Realizar una consulta con los datos referentes a la colisión a eliminar.	
	3	Pulsar el botón derecho sobre la fila a eliminar. Esta se seleccionará entera y aparecerá un menú con la opción "Eliminar fila".	
	4	Seleccionar dicha opción y esa fila desaparece de la pantalla y de la base de datos.	
Poscondición	La colisión quedará eliminada.		
Excepciones	Paso	Acción	
	3	Si la consulta realizada en el paso 2 no devuelve resultados, dicho vehículo ya no existe.	
Comentarios			

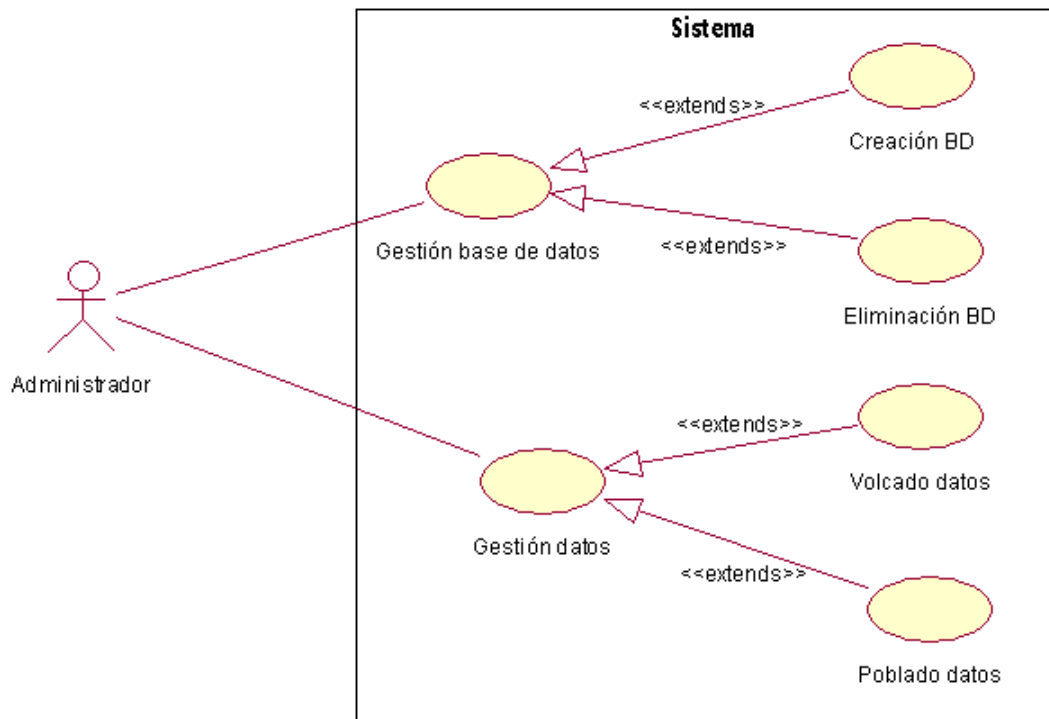


Figura 2. Diagrama de casos de uso del programa administrador.

## Definición de actores

ACT-02	Administrador
<b>Descripción</b>	Representa al usuario administrador de la base de datos. Usuario con conocimiento de la base de datos y su contenido.
<b>Comentarios</b>	Total acceso a todas las partes del programa. Tiene la responsabilidad de la seguridad en la base de datos, de la creación de copias de seguridad de los datos, etc.

## Casos de uso del sistema

Igual que en los casos de uso referidos al programa principal procedemos a analizar la funcionalidad del programa administrador de la base de datos mediante los casos de uso que vamos a exponer a continuación, consiguiendo representar la funcionalidad de cada uno de sus componentes y sus relaciones.

<b>RF-09</b>	<b>Creación BD</b>	
<b>Objetivos asociados</b>	OBJ-04 Administración de la base de datos	
<b>Requisitos asociados</b>	RNF-03 Control de la base de datos y su contenido	
<b>Descripción</b>	Cuando el usuario desee crear las tablas o relaciones de la base de datos necesaria para el uso del programa principal de gestión de accidentes.	
<b>Precondición</b>	La base de datos no debe estar creada.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario selecciona la pestaña "Gestión base de datos"
	<b>2</b>	Pincha en "Crear Base de Datos"
	<b>3</b>	El sistema comprueba que la base de datos no exista. Si es así, la crea.
<b>Poscondición</b>	Se ha creado la base de datos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si por el contrario esta ya existe, se informa de ello al usuario mediante un pantallazo y se cancela la creación.
<b>Comentarios</b>		

<b>RF-10</b>	<b>Eliminación BD</b>	
<b>Objetivos asociados</b>	OBJ-04 Administración de la base de datos	
<b>Requisitos asociados</b>	RNF-03 Control de la base de datos y su contenido	
<b>Descripción</b>	Cuando el usuario desee eliminar las tablas o relaciones de la base de datos así como todo su contenido.	
<b>Precondición</b>	La base de datos debe estar creada.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario selecciona la pestaña "Gestión base de datos"
	<b>2</b>	Pincha en "Eliminar Base de Datos"
	<b>3</b>	El sistema comprueba que la base de datos exista. Si es así, muestra una pantalla de seguridad.
	<b>4</b>	Dicha pantalla te informa de lo que vas a hacer, si aceptas la base de datos es eliminada.
<b>Poscondición</b>	Se ha eliminado la base de datos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si por el contrario la base de datos no existe, se informa de ello al usuario

		mediante un pantallazo y se cancela la eliminación.
	<b>4</b>	Si no aceptas en la pantalla de seguridad, la eliminación queda cancelada.
<b>Comentarios</b>	Al eliminar la base de datos, su contenido desaparece. Por ello se aconseja realizar primero un volcado de los datos.	

<b>RF-11</b>	<b>Volcado datos</b>	
<b>Objetivos asociados</b>	OBJ-05 Control del contenido de la base de datos	
<b>Requisitos asociados</b>	RNF-03 Control de la base de datos y su contenido	
<b>Descripción</b>	Cuando el usuario desee almacenar en ficheros de texto los datos almacenados en la base de datos.	
<b>Precondición</b>	La base de datos debe estar creada.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario selecciona la pestaña "Control de datos"
	<b>2</b>	Pincha en "Volcado de datos"
	<b>3</b>	El sistema comprueba que la base de datos exista. Si es así, muestra una pantalla con todos los datos almacenados.
	<b>4</b>	Utilizamos la opción volcar datos. Generando un fichero de extensión ".txt" por cada tabla o relación existente en la base de datos.
<b>Poscondición</b>	Se ha eliminado la base de datos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si por el contrario la base de datos no existe, se informa de ello al usuario mediante un pantallazo y se cancela el volcado.
<b>Comentarios</b>	El volcado de datos se hace con un formato en concreto que facilite la vuelta de los datos a la base de datos, en caso de que estos sean borrados.	

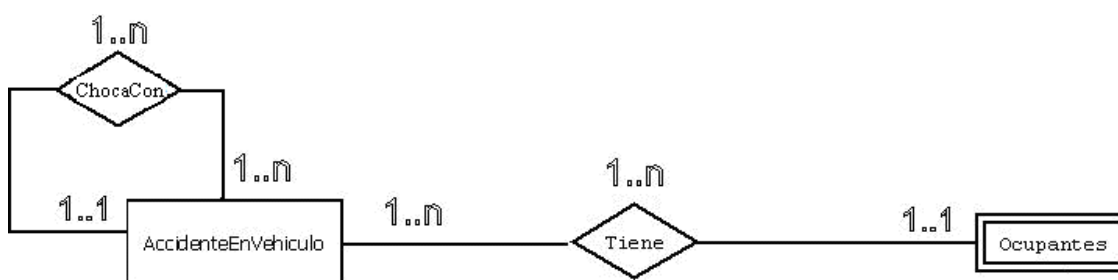
<b>RF-12</b>	<b>Poblado datos</b>	
<b>Objetivos asociados</b>	OBJ-05 Control del contenido de la base de datos	
<b>Requisitos asociados</b>	RNF-03 Control de la base de datos y su contenido	

<b>Descripción</b>	Cuando el usuario desee almacenar los datos guardados en ficheros, con la opción volcar datos, en la base de datos.	
<b>Precondición</b>	La base de datos debe estar creada y los datos no deben estar ya almacenados.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	<b>1</b>	El usuario selecciona la pestaña "Control de datos"
	<b>2</b>	Pincha en "Poblado de datos"
	<b>3</b>	El sistema comprueba que la base de datos exista. Si es así, muestra una pantalla con todos los datos almacenados. Entre los que encontramos los almacenados en los ficheros.
	<b>4</b>	Unos pantallazos nos muestran si el poblado es correcto o erróneo.
<b>Poscondición</b>	Se ha poblado la base de datos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	<b>3</b>	Si alguno de los datos ya existe en la base de datos, se cancelará su inserción.
<b>Comentarios</b>		

## 2.5 - Diagrama Entidad-Relación

Para la realización de la herramienta es necesario el almacenamiento de toda la información de forma permanente, por ello se va a utilizar una base de datos donde se almacenará toda la información necesaria.

Para la creación de dicha base de datos, previamente se ha diseñado un diagrama entidad-relación, que podemos ver en la siguiente figura.



## **Capítulo 3**

# **DISEÑO**

## 3.1 – Diseño relacional de la base de datos

A partir del diagrama entidad/Relación descrita anteriormente se ha realizado el diseño relacional de la base de datos. Para ello, partiendo de las entidades definidas se ha llevado a cabo el paso a tablas, en el que cada entidad anterior ha pasado a ser una tabla. Los campos de cada una vienen determinados por los atributos definidos en el diseño anterior. Una vez diseñadas las tablas principales, se ha pasado a diseñar las relaciones entre las mismas. Este proceso genera el siguiente resultado:

### VehiculoAccidentado

En esta entidad se va a almacenar toda la información referente a los vehículos que han sufrido un accidente. Los atributos a almacenar son los siguientes:

- **Matricula:** Identificador inequívoco de cada uno de los automóviles, correspondiente al propio número de la placa de la matrícula de este. Este atributo forma parte de la clave primaria de esta tabla.
- **Fecha:** Indica el año, mes y día en que el vehículo, identificado por su matrícula, sufrió el accidente. También forma parte de la clave primaria.
- **Hora:** Momento en el que se produjo el accidente, expresado en horas y minutos. Último atributo de la clave primaria.
- **NumOcupantes:** Número de ocupantes que circulaban en el vehículo en el momento del accidente.
- **TipoColision:** Almacena la forma en que el vehículo colisionó, en que de existir colisión, siendo las posibilidades: frontal, lateral, alcance, despeñamiento y vuelco sin colisión.
- **VelocidadExcesiva:** Atributo booleano que indica si el vehículo llevaba exceso de velocidad en el momento del accidente. Guiándonos para valorarlo en las normas de tráfico referentes al tipo de vía por la que circulaba.
- **TipoVehiculo:** Engloba el tipo de vehículo entre unas opciones definidas de antemano: turismo pequeño, turismo grande, todoterreno, furgoneta, camión.
- **VelocidadEstimada:** Estimación realizada a posteriori de la velocidad a la que circulaba el vehículo. También se permite la opción "Se desconoce" en caso de no poder estimarla.
- **Airbag:** Atributo booleano que indica si el vehículo contaba o no con airbag.

- **EuronCAP:** Este atributo almacena las estrellas que se le han dado al vehículo. Para entenderlo definiremos EuronCAP como una serie de pruebas de seguridad pasiva realizadas en automóviles nuevos entregando una clasificación en estrellas basada en el comportamiento del vehículo en pruebas de impacto frontal y lateral.

## Ocupantes

En esta entidad se almacenará de forma detallada los datos y la valoración médica, tanto en el lugar del accidente como en el hospital, de los ocupantes de los vehículos accidentados. Para ello necesitaremos los siguientes atributos:

- **Matricula:** Clave foránea perteneciente a la entidad "VehiculoAccidentado". Necesario para saber en que coche circulaba cada uno de los ocupantes.
- **Fecha:** Ídem que el atributo anterior, es necesario para identificar el vehículo.
- **Hora:** Ídem que los dos anteriores.
- **Id:** Identificador interno de la aplicación necesario para la identificación unívoca de cada uno de los diferentes ocupantes de los vehículos.
- **Edad:** Años con los que contaba cada ocupante en concreto en el momento del accidente.
- **IDE:** Atributo a destacar puesto que su calculo es el objetivo de la tesis para la que estoy desarrollando esta aplicación. Su significado es Identificador de Deformación Estructural e indica la posible gravedad de cada ocupante en función de las deformaciones producidas en el vehículo con razón del accidente.
- **Glasgow:** Atributo de carácter numérico que se utiliza para medir el nivel de conciencia del paciente. Incluimos en la base de datos tres mediciones de este campo:
  - **Glasgow Samu:** tomado por los sanitarios de las ambulancias.
  - **Glasgow Urg:** medido en urgencias hospitalarias.
  - **Glasgow Uci:** medido en la unidad de cuidados intensivos.
- **RTS:** Revised trauma score. Atributo numérico que refleja el índice de gravedad basado en criterios fisiológicos (frecuencia respiratoria, tensión arterial sistólica y Glasgow). Incluimos en la base de datos tres mediciones de este campo:
  - **RTS Samu:** tomado por los sanitarios de las ambulancias.
  - **RTS Urg:** medido en urgencias hospitalarias.
  - **RTS Uci:** medido en la unidad de cuidados intensivos.



- **ISS:** Injury Severity Score. Atributo numérico que valora la gravedad atendiendo a criterios anatómicos, es decir dependiendo de las zonas que el paciente presenta lesionadas.  
Incluimos en la base de datos dos mediciones de este campo:
  - **ISS Urg:** medido en urgencias hospitalarias.
  - **ISS Uci:** medido en la unidad de cuidados intensivos.
- **LesiónPrincipal:** Atributo que indica cual es la lesión de mayor gravedad del accidentado, de mucha utilidad a la hora de tratar al paciente.
- **TipoAtención:** Dependiendo de la gravedad de las heridas sufridas por cada ocupante este recibirá uno de los tres tipos de atención: forense, UCI o NoEtna.
- **Tce:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo craneoencefálico.
- **Tfacial:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo facial.
- **Ttraquis:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo medular.
- **Ttorax:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo en el torácico.
- **Tabd:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo abdominal.
- **Tee:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo en las extremidades.
- **Tbland:** Atributo booleano que indica si el accidentado sufrió o no un traumatismo en los tejidos blandos (heridas, laceraciones...).

## ChocaCon:

Entidad utilizada para el almacenamiento de toda la información necesaria de las colisiones entre vehículos. Los atributos que la forman son:

- **Matricula:** Clave foránea perteneciente a la entidad "VehiculoAccidentado". Necesario para conocer todas las características del vehículo y de sus ocupantes en el momento de la colisión.
- **Fecha:** Ídem que el atributo anterior, es necesario para identificar el vehículo y sus ocupantes.
- **Hora:** Ídem que los dos anteriores.
- **Matricula2:** Clave foránea perteneciente a la entidad "VehiculoAccidentado". Necesario para conocer todas las

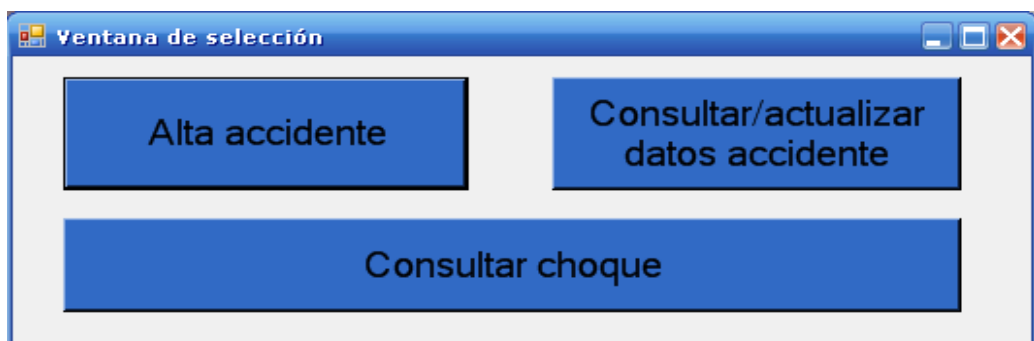
características del resto de los vehículos implicados en la colisión, así como todos los datos de los ocupantes.

## 3.2 – Diseño de las interfaces

A la hora de diseñar las interfaces gráficas para las aplicaciones también hay que tener en cuenta una serie de criterios. Deben ser unas interfaces lo más sencillas posibles para que el usuario pueda comprenderlas sin dificultad. También deben tener visibles en todo momento las opciones disponibles, y la navegación por todas ellas debe ser rápida e intuitiva. Estas opciones deben tener nombres lo más descriptivos posibles para que con sólo leerlos se pueda saber qué funcionalidad ofrece. Los colores utilizados deben ser suaves para conseguir así que resulte lo más agradable posible.

### Programa principal:

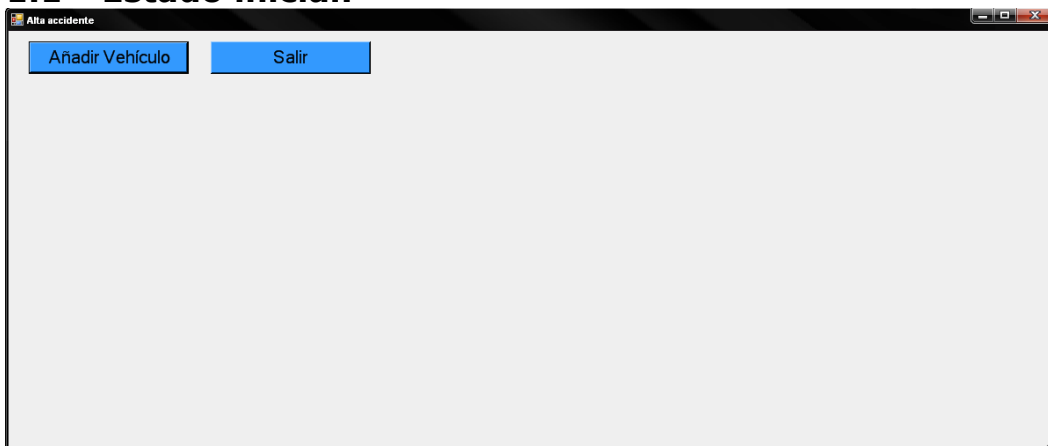
En la parte superior derecha de la pantalla aparecen los tres botones típicos de manejo de las pantallas: minimizar, maximizar y cerrar. Necesarios para las funcionalidades mínimas que debe tener cualquier tipo de interfaz.



Dentro de la ventana de la interfaz encontramos tres botones que dan acceso a las utilidades de la aplicación:

- 1- **Alta accidente:** muestra otra pantalla en la que tendremos la posibilidad de introducir a la base de datos la información de un nuevo accidente, compuesta por los datos del vehículo y los de los ocupantes que viajaban en él.

## 1.1- Estado inicial:



La pantalla aparece únicamente con dos botones “Añadir Vehículo” y “Salir”:

- Con el primero accedemos al apartado diseñado para la inserción de los datos del vehículo. Una vez los datos sean introducidos correctamente en el sistema, este nos posibilitará las opciones para introducir sus ocupantes.
- Pinchando en el segundo botón saldremos de esta ventana, volviendo a la ventana inicial.

## 1.2- Ejemplo de alta de un vehículo y sus ocupantes:

A screenshot of the 'Alta accidente' window showing a form for entering accident data. The form is divided into two main sections. The top section contains a table with fields for vehicle information: Matricula (1250CWC), Fecha (2010-05-31), Hora (12:00), NumOcupantes (2), TipoColision (Alcance), VelocidadExcesiva (Desconocida), TipoVehiculo (Turismo(pequeño)), VelocidadEstimada (0), Airbag (Si), EuronCAP (5), and AñoMatriculacion (Entre 5 y 10 años). Below this is a blue button labeled 'Añadir Ocupante'. The bottom section contains a table for occupant information with columns: Edad (23), IDE (3), Glasgowamu (3), Glasgowurg (1), Glasgowuci (1), RTSamu (3), RTSurg (1), RTSuci (2), ISSug (3), ISSuci (3), LesionPrincipal (Traumatismo craneoencefálico), TipoAtencion (UCI), Tce (Si), Tfacial (No), and T (N). At the bottom right of the form are two blue buttons: 'Cancelar' and 'Guardar'.

La aplicación muestra los datos introducidos y posibilita realizar cambios en ellos. La funcionalidad de los botones que observamos es la siguiente:

- **Añadir ocupante:** nos permite introducir los datos de uno de los ocupantes del vehículo en el momento del accidente, relacionándolo como tal en la base de datos. La cota máxima de ocupantes la marca la variable “NumOcupantes”, por lo que una vez que lleguemos a su valor este botón quedará deshabilitado.

- **Cancelar:** Nos permite salir de la pantalla actual sin guardar la información que hemos añadido, volviendo a la ventana inicial de la aplicación.
- **Guardar:** Almacena en la base de datos todo lo que se muestra en la pantalla. Después la cierra y vuelve a la ventana inicial.

**2- Consultar/actualizar datos accidente:** nueva pantalla que nos permite hacer todo tipo de consultas sobre la base de datos, además de facilitarnos la opción de cambiar o eliminar la información almacenada tanto de los vehículos como de sus ocupantes. También esta pantalla nos muestra la opción de "Añadir colisión" necesaria para poder almacenar el choque entre diversos vehículos, en caso de ser esa la causa del accidente por el que están almacenados.

### 2.1- Estado inicial:

La aplicación nos muestra tres cuadros de diálogo y dos botones:

- **Cuadros de diálogo:** son matricula, fecha y hora. Las tres variables sobre las que haremos las consultas. Todos los cuadros pueden rellenarse desde el teclado o desplegando la "dropDownList" que nos ofrece una muestra de los posibles valores que puede tomar cada una de las variables en función de lo que haya almacenado en la base de datos. Al menos uno de los cuadros debe estar rellenado para poder realizar una consulta
- **Botones:** El botón "Consultar" realiza la consulta sobre la base de datos en función de los datos introducidos en los cuadros de diálogo y muestra sus resultados. Con el botón "Salir" saldremos de esta ventana, volviendo a la ventana inicial.

## 2.2- Ejemplo de consulta:

**Muestra/actualiza accidentee**

Matricula:  Fecha: 2010-05-31 Hora:

*NOTA: Para eliminar una linea haz click con el boton derecho sobre ella*

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
1250JCWC	31/05/2010	12:00	2	Alcance	Desconocida	Turismo(pequeño)	0	Si	5	Entre 5 y 10 años

Edad	IDE	Glasgowamu	Glasgowwug	Glasgowuci	RTSxamu	RTSug	RTSuci	ISSug	ISSuci	LesionPrincipal	TipoAtencion	Tce	Tfacial	T
23	3	3	1	1	3	1	2	3	3	Traumatismo craneoencefálico	UCI	Si	No	N

Los botones del estado inicial se mantienen, posibilitando la realización de otra consulta en cualquier momento o la salida de esta pantalla.

Para mostrar los datos de los ocupantes relacionados con cada vehículo mostrado, haremos clic sobre cualquiera de los datos del vehículo. Todos los datos mostrados se pueden modificar utilizando los "dropDownList" (en las variables que lo permitan) o escribiendo desde el teclado. Los cambios realizados son almacenados directamente en la base de datos.

También podemos eliminar vehículos y ocupantes usando el botón derecho, tal y como indica la "NOTA" que podemos observar debajo de los cuadros de dialogo.

En casos concretos pueden aparecer dos nuevos botones:

- **Añadir ocupante:** La aplicación lo muestra únicamente cuando el número de ocupantes almacenados, relacionados con el vehículo seleccionado en ese momento, es inferior al valor de la variable "NumOcupantes". Una vez que dicho valor sea igual este, el botón se deshabilitará.
- **Añadir colisión:** mostrado por la aplicación cuando la consulta requerida muestra al menos dos vehículos. Nos permite almacenar la colisión entre dos vehículos siempre y cuando ambos tengan iguales los valores de las variables fecha y hora.

**3- Consultar choque:** pantalla que realiza consultas sobre la base de datos buscando si un vehículo concreto tuvo una colisión con otro vehículo, utilizando unos datos temporales específicos e

introducidos por el usuario. También nos permite eliminar dichas colisiones de la base de datos.

### 3.1- Estado inicial:

Matricula:  Fecha:  Hora:

NOTA: Para eliminar una linea haz click con el boton derecho sobre ella

Resto de vehículos implicados:

Ídem que el estado inicial de "**Consultar/actualizar datos accidente**" pero sabiendo que en esta pantalla las consultas son de colisiones entre vehículos concretos en una fecha específica, por lo que el sistema exige que se rellenen los tres cuadros de diálogo.

### 3.2- Ejemplo de colisión:

Matricula:  Fecha:  Hora:

NOTA: Para eliminar una linea haz click con el boton derecho sobre ella

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
2	27/05/2010	11:22	5	Lateral	No		0	No	0	

Edad	IDE	GlasgowSamu	Glasgowwurg	Glasgowwuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	lesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
22	2	2	2	2	2	2	2			UCI	No	No	No	No
2	1		1	1	1					adidos piena...	UCI	No	No	No

Resto de vehículos implicados:

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
34	27/05/2010	11:22	1	Vuelco sin colisión	No		0	No	0	

Edad	IDE	GlasgowSamu	Glasgowwurg	Glasgowwuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	lesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
21	2	2	2	2	2	2	2			Forense	No	No	No	No

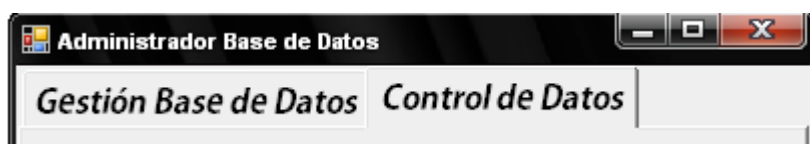
Para mostrar los datos de los ocupantes de cualquiera del "Resto de vehículos implicados" haremos clic sobre alguno de los datos del vehículo concreto.

Las colisiones mostradas pueden ser eliminadas usando el botón derecho, tal y como muestra la "NOTA" mostrada debajo de los cuadros de diálogo. Debemos tener en cuenta que únicamente podemos eliminar colisiones por lo que los datos de los ocupantes no podrán ser eliminados.

Esta pantalla no de la interfaz no permite la opción de actualización.

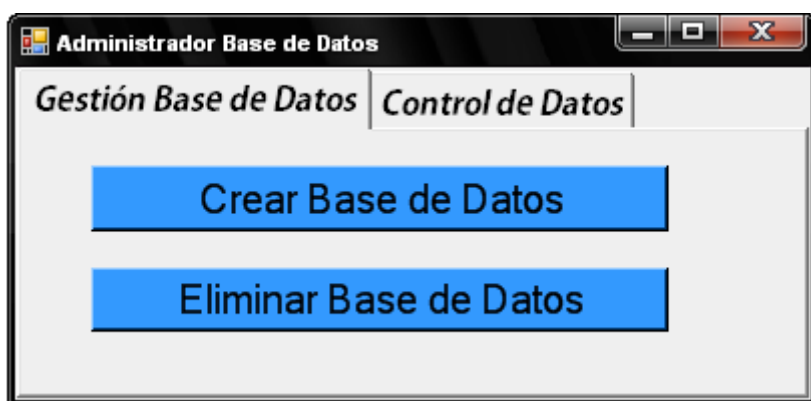
## Administrador:

Al igual que el programa anterior este posee en la parte superior derecha de la pantalla los tres botones típicos de manejo de las pantallas: minimizar, maximizar y cerrar. Necesarios para las funcionalidades mínimas que debe tener cualquier tipo de interfaz.



En su interior encontramos dos pestañas que separan la aplicación en dos partes, la que hace referencia a la base de datos y la que se encarga de su contenido:

### 1- Gestión Base de Datos:

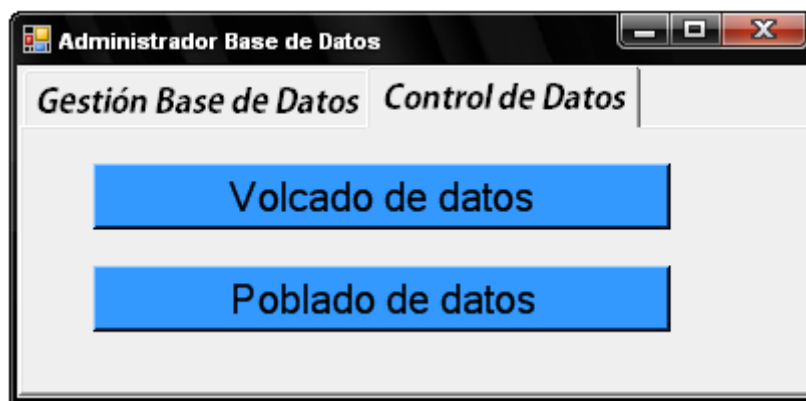


Sus funcionalidades principales, expresadas con los dos botones que se muestran en esta pantalla, son:

**1.1- Crear Base de Datos:** Crea todas las tablas o relaciones necesarias para que el programa principal interactúe sobre ellas de forma segura.

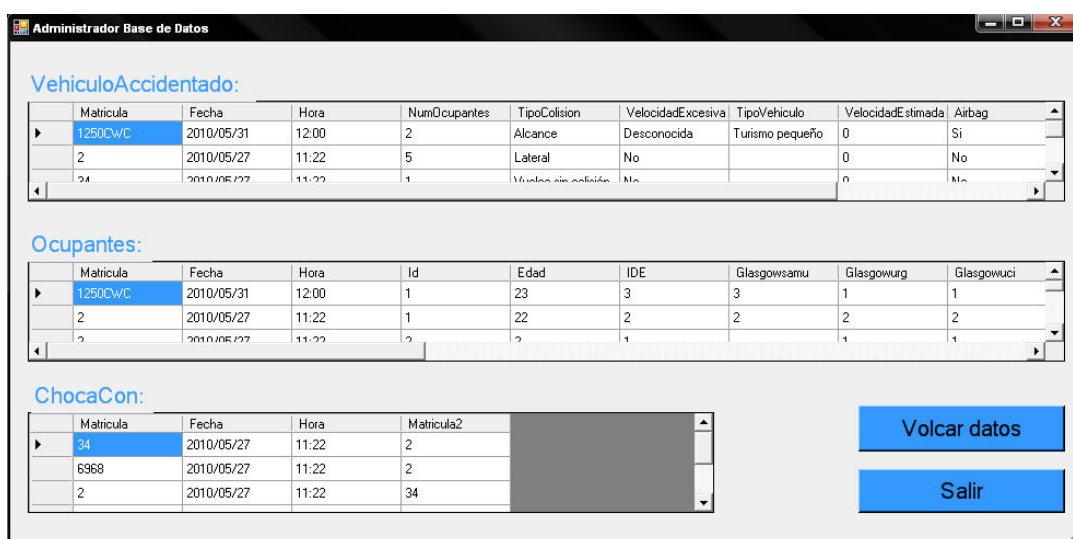
**1.2- Eliminar Base de Datos:** Elimina todo lo creado en el apartado 1.1 y por consiguiente todos los datos almacenados en la base de datos. Al considerarse una acción de alto riesgo al clicar en este botón saltará una venta de seguridad informándonos de lo que vamos ha hacer y posibilitándonos el interrumpir la acción.

## 2- Control de datos:



Al igual que explico en el apartado 1 las funcionalidades principales de esta pantalla están expresadas en los dos botones que presenta en su interior:

**2.1- Volcado de datos:** Almacena en ficheros de texto toda la información contenida en la base de datos. Guardándola con un formato concreto para facilitar su posterior inserción. Al hacer clic sobre este botón el sistema nos proporciona una visión general de los datos guardados:



**VehiculoAccidentado:**

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag
1250CW/C	2010/05/31	12:00	2	Alcance	Desconocida	Turismo pequeño	0	Si
2	2010/05/27	11:22	5	Lateral	No		0	No
24	2010/05/27	11:22	1	Vehículo sin control	No		0	No

**Ocupantes:**

Matricula	Fecha	Hora	Id	Edad	IDE	Glasgowsamu	Glasgowuig	Glasgowuci
1250CW/C	2010/05/31	12:00	1	23	3	3	1	1
2	2010/05/27	11:22	1	22	2	2	2	2
2	2010/05/27	11:22	2	2	1		1	1

**ChocaCon:**

Matricula	Fecha	Hora	Matricula2
34	2010/05/27	11:22	2
6968	2010/05/27	11:22	2
2	2010/05/27	11:22	34

**Volcar datos**

**Salir**

**2.2- Poblado de datos:** Al utilizar esta opción el sistema utiliza los ficheros de texto preparados en el apartado 2.1 para rellenar la base de datos con todos los datos anteriores. Para mostrar el correcto poblado el sistema muestra una ventana con todos los datos de la base de datos y mediante pantallazas te indica el correcto poblado de las diferentes tablas o relaciones:



Administrador Base de Datos

**VehiculoAccidentado:**

	Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag
▶	1212	2010/03/06	12:12	3	Frontal	Desconocida	Todoterreno	0	No
	123	2010/03/18	12:00	1	Frontal	No	Turismo pequeño	128	No
	1250CWC	2010/05/31	12:00	2	Alarma	Desconocida	Turismo pequeño	0	Si

**Ocupantes:**

	Matricula	Fecha	Hora	Id	Edad	IDE	Glasgowscmu	Glasgowurg	Glasgowuci
▶	1250CWC	2010/05/31	12:00	1	23	3	3	1	1
	2	2010/05/27	11:22	1	22	2	2	2	2
	3	2010/05/27	11:22	2	2	1	1	1	1

**ChocaCon:**

	Matricula	Fecha	Hora	Matricula2
▶	34	2010/05/27	11:22	2
	6968	2010/05/27	11:22	2
	43	2010/03/18	12:21	3

Poblado correcto de la tabla VehiculoAccidentado.

Aceptar

**Salir**

# **Capítulo 4**

## **IMPLEMENTACIÓN**

## 4.1 - Herramientas utilizadas (Visual Studio .NET, MySQL Server 6.0)

El sistema operativo usado por los clientes para los que desarrollo la aplicación es Windows, por lo que todo el desarrollo se realizará en Visual Studio .NET. Para el almacenamiento de los datos nos valdremos de MySQL Server 6.0. Ya que, como expliqué en los requisitos hardware, es un programa GNU, gratuito (si no se utiliza para productos privativos en los que la empresa si debería comprar una licencia específica) y que posee las librerías necesarias para trabajar sobre él desde Visual Studio .NET.

Las razones por las que elegí estas herramientas son las siguientes:

- En primer lugar el Visual Studio .NET es un entorno de desarrollo integrado para sistemas operativos Windows que soporta varios lenguajes de programación tales como Visual C++ [VC++], Visual C# [VC#], Visual J# [VJ#], ASP.NET [Asp.Net] y Visual Basic .NET [Vb.Net]. Visual Studio permite a los desarrolladores crear aplicaciones, sitios, aplicaciones Web, así como servicios Web en cualquier entorno que soporte la plataforma .NET (a partir de la versión Net 2002).
- En cuanto al lenguaje de programación, se ha elegido Visual Basic .NET, orientado a objetos e implementada sobre el Framework .NET [Frw.Net]. El principal motivo de esta decisión ha sido el alto grado de conocimiento y familiarización tanto con el entorno de desarrollo elegido como con el lenguaje.
- Como también se ha comentado en la parte de requisitos, necesitamos para el almacenamiento de los datos una alta disponibilidad y un servidor de base de datos robusto. Además, uno donde el volumen de datos no sea un problema, ya que se espera que el tamaño vaya aumentando de forma considerable, así como el tiempo de acceso a los mismos no sea lento ya que el trabajo que se realiza es en tiempo real y el tiempo de respuesta es fundamental. Por esto y por lo explicado en el primer párrafo se optó por el uso de MySQL Server 6.0.

Tras haber desarrollado la aplicación considero acertada la decisión de las herramientas. El entorno de trabajo me ha facilitado mucho el desarrollo de la interfaz, el lenguaje orientado a objetos me ha proveído de técnicas avanzadas necesarias para una mejor distribución del código (herencias, polimorfismos, etc.) y el servidor de bases de datos ha cumplido con creces su objetivo.

## 4.2 - Incidencias y soluciones aplicadas

Los problemas encontrados a la hora de desarrollar la aplicación han sido muchos y su solución ha implicado diferentes grados de dificultad. En estas páginas mostrare algunos ejemplos:

### 1- Introducir valores correctos en los formulario

#### Problema

Cada uno de los campos de diálogo mostrados en los formularios debe ser rellenado con un tipo de datos concreto, si se incumple, la inserción en la base de datos resultará errónea y, por lo tanto, se perderán los datos introducidos. También existen campos cuyo contenido puede ser vacío.

#### Solución

Las soluciones a este problema son varias, ya que un mismo problema en distintos sitios debe ser tratado de maneras diferentes:

- Campos en blanco:  
Los valores en blanco únicamente son aceptados en la base de datos para variables de tipo String. Por ello las variables de tipo smallint (int16 en Visual Basic) no pueden ser vacío:
  - En los casos como EuronCap o VelocidadEstimada, si el usuario no rellena estos campos introduzco un cero en la base de datos en vez de vacío.

```
'Ejemplo con la variable VelocidadEstimada
If IsDBNull (Me.DataGridView1.Rows (linea).Cells(7).Value()) Then
    Me.DataGridView1.Rows (linea).Cells(7).Value() = 0
End If
```

- En variables médicas de carácter numérico, como por ejemplo RTS (cualquiera de las tres) el valor cero esta dentro de la escala que lo mide y esta medida no es obligatorio que se le realice a todos los pacientes, lo que posibilita la opción de dejar vacío este campo. La solución ha sido cambiar en la base de datos el tipo de variable por String, y comprobar en todos los casos que es un entero antes de insertar ningún valor en la base de datos.

```

'Comprueba que los valores desde Edad hasta ISSuci sean de tipo
entero
Dim nuevoValor As String

If IsDBNull(Me.DataGridView2.Rows(e.RowIndex).Cells(1).Value()) Then
    MessageBox.Show("La edad es un campo obligatorio y no puede ser
vacio.")
    Me.DataGridView2.Rows(e.RowIndex).Cells(1).Value() =valorEnteros
End If

If IsDBNull(Me.DataGridView2.Rows(e.RowIndex).Cells(2).Value()) Then
    Me.DataGridView2.Rows(e.RowIndex).Cells(2).Value() =valorEnteros
    MessageBox.Show("El IDE es un campo obligatorio y no puede ser
vacio.")
End If

If (e.ColumnIndex < 11) Then
    Try
        If IsDBNull(Me.DataGridView2.Rows(e.RowIndex).Cells
(e.ColumnIndex).Value()) Then
            Me.DataGridView2.Rows(e.RowIndex).Cells(e.ColumnIndex
).Value() = ""
        Else
            nuevoValor = Me.DataGridView2.Rows(e.RowIndex).Cells
(e.ColumnIndex).Value()
            Dim esEntero As Integer = CType(nuevoValor, Int16)
        End If
    Catch ex As OverflowException
        MessageBox.Show("El valor de " & Me.DataGridView2.Columns
(e.ColumnIndex).Name & " de la columna " & (e.RowIndex +
1) & " es demasiado grande.")

        Me.DataGridView2.Rows(e.RowIndex).Cells(e.ColumnIndex).Valu
e() = valorEnteros
    Catch ex As Exception
        MessageBox.Show("Debe ser numérico el valor de " &
Me.DataGridView2.Columns(e.ColumnIndex).Name & " de la
columna " & (e.RowIndex + 1) & ".")

        Me.DataGridView2.Rows(e.RowIndex).Cells(e.ColumnIndex).Valu
e() = valorEnteros
    End Try
End If

```

- Tipos de datos concretos:

Cuando el usuario introduce al sistema los datos de un accidente en un vehículo o de un ocupante lo hace mediante unos formularios con explicaciones de los tipos de datos que debe introducir en cada campo, y las pulsaciones de las teclas están controladas, es decir, en un campo numérico únicamente se permiten las pulsaciones sobre teclas numéricas. Aun así, si el usuario incumple alguna de las instrucciones el sistema se lo notificará mediante un pantallazo.

```

'Ejemplo de control de las pulsaciones para la variable Hora
Private Sub Hora_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles Hora.KeyPress

```

```

    If (Not IsNumeric(e.KeyChar)) And (e.KeyChar <> ChrW(Keys.Back))
    And (e.KeyChar <> ":") Then
        e.Handled = True
    End If
End Sub

'Función que comprueba los posibles errores en la inserción de los
datos de un vehículo
Public Function compruebaErrores() As Integer
    Dim primerGuion, segundoGuion, dosPuntos, longMatricula,
    longOcupantes, pruebaEntero, longHora, horas, minutos As Integer
    Dim textoFecha, textoHora, textoMatricula, Ocupantes, aux As
    String
    Dim hayErrores As Integer = 0

    muestraErrores = ""
    'Controlo que la fecha este correcta
    textoFecha = Me.Fecha.Text
    primerGuion = textoFecha.IndexOf("-")
    segundoGuion = textoFecha.LastIndexOf("-")
    If (primerGuion <> 4) And (segundoGuion <> 7) Then
        muestraErrores = muestraErrores & "* El formato de la fecha
        debe ser XXXX-XX-XX" & vbCrLf
        hayErrores = 1
    End If

    Try
        'Controlo la correccion en el formato de la hora
        textoHora = Me.Hora.Text
        longHora = textoHora.Length()
        If (longHora = 0) Then
            muestraErrores = muestraErrores & "* El formato de la
            hora debe ser XX:XX (24horas)" & vbCrLf
            hayErrores = 1
        Else
            dosPuntos = textoHora.IndexOf(":")
            Try
                minutos = CType(textoHora.Substring(3), Integer)
                aux = textoHora.Remove(2, 3)
                horas = CType(aux, Integer)
            Catch ex As Exception
                muestraErrores = muestraErrores & "* Caracteres
                ilegales en la hora" & vbCrLf
                hayErrores = 1
            End Try
            If (dosPuntos = -1) Or (longHora <> 5) Or (minutos >
            59) Or (horas > 23) Then
                muestraErrores = muestraErrores & "* El formato de
                la hora debe ser XX:XX (24horas)" & vbCrLf
                hayErrores = 1
            End If
        End If
    End If

    Catch ex As Exception
    End Try

    'Controlo que el campo de la matricula no este en blanco
    textoMatricula = Me.Matricula.Text
    longMatricula = textoMatricula.Length()

```

```

If (longMatricula = 0) Then
    muestraErrores = muestraErrores & "* Debes introducir la
matricula del vehiculo" & vbCrLf
    hayErrores = 1
End If

'Controlo el carater numerico de la expresion del numero de
ocupantes
Ocupantes = Me.NumOcupantes.Text
longOcupantes = Ocupantes.Length()

If (longOcupantes = 0) Then
    muestraErrores = muestraErrores & "* Debes introducir el
número de ocupantes del vehiculo" & vbCrLf
    hayErrores = 1
End If

Try
    pruebaEntero = CType(Ocupantes, Int16)
Catch ex As Exception
    muestraErrores = muestraErrores & "* El número de ocupantes
debe ser un entero" & vbCrLf
    hayErrores = 1
End Try

Return hayErrores
End Function

```

## 2- Actualización valores

### Problema

Este problema se presenta en dos partes de la aplicación: "Alta accidente" y "Consultar/actualizar datos accidente", y se basa en la dificultad para controlar los caracteres introducidos en los `DataGridView` (herramienta elegida para mostrar de forma ordenada los datos introducidos por el usuario). El control del tipo de valores se realiza de forma parecido a la explicada en el apartado uno, pero se nos presenta otro problema con algunas variables que deben tomar valores concretos de una lista.

El segundo gran problema se presenta en la actualización del valor de la variable "NumOcupantes" (representa al número de ocupantes que viajaban en el vehículo):

- Si introducimos un nuevo valor que sea menor del número de ocupantes ya guardado en el sistema, debe producirse un error.
- Si introducimos un valor igual al número de ocupantes ya guardado en el sistema, se debe deshabilitar la opción de añadir nuevos ocupantes.
- Si introducimos un valor que sea mayor al número de ocupantes ya guardado en el sistema, se debe habilitar la opción de añadir nuevos ocupantes.

## Solución

Para aquellas variables que deben tomar su valor de una lista implemento una herramienta que añadiré a las DataGridView, las llamadas `DataGridViewComboBoxCell()`. Que posibilitan añadir una lista desplegable a celdas concretas.

```
'Añado los comboBox necesarios al dg1
Public Sub comboDg1()
    Try
        'Cambiando a comboBox las celdas necesarias
        For index As Integer = 0 To DataGridView1.RowCount
            Dim datosColision As New DataGridViewComboBoxCell()
            Dim datosTipoVehiculo As New DataGridViewComboBoxCell()
            Dim datosMatriculacion As New DataGridViewComboBoxCell()
            Dim datosVelocidadEx As New DataGridViewComboBoxCell()
            Dim datosAirbag As New DataGridViewComboBoxCell()

            datosColision.Items.Add("Frontal")
            datosColision.Items.Add("Lateral")
            datosColision.Items.Add("Alcance")
            datosColision.Items.Add("Vuelco sin colisión")
            datosColision.Items.Add("Despeñamiento")

            datosTipoVehiculo.Items.Add("Turismo (pequeño)")
            datosTipoVehiculo.Items.Add("Turismo (grande)")
            datosTipoVehiculo.Items.Add("Todoterreno")
            datosTipoVehiculo.Items.Add("Furgoneta")
            datosTipoVehiculo.Items.Add("Camión")

            datosVelocidadEx.Items.Add("Si")
            datosVelocidadEx.Items.Add("No")
            datosVelocidadEx.Items.Add("Desconocida")

            datosAirbag.Items.Add("Si")
            datosAirbag.Items.Add("No")

            datosMatriculacion.Items.Add("Más de 10 años")
            datosMatriculacion.Items.Add("Entre 5 y 10 años")
            datosMatriculacion.Items.Add("Menos de 5 años")

            DataGridView1.Item(4, index) = datosColision
            DataGridView1.Item(5, index) = datosVelocidadEx
            DataGridView1.Item(6, index) = datosTipoVehiculo
            DataGridView1.Item(8, index) = datosAirbag
            DataGridView1.Item(10, index) = datosMatriculacion
        Next
    Catch ex As Exception
    End Try
End Sub
```

Para el problema de los ocupantes preparé dos funciones que se ejecutan antes y después de la actualización de cualquier variable del DataGridView pero que únicamente trataran a celdas concretas:



## ANTES:

```
Private Sub DataGridView1_CellBeginEdit(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellCancelEventArgs) Handles DataGridView1.CellBeginEdit
    Dim linea As Integer

    linea = e.RowIndex
    numOcupantesHay = Me.DataGridView1.Rows(linea).Cells(4).Value()
End Sub
```

## DESPUÉS:

```
Public Sub ActualizaDg1(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
    Dim linea As Integer
    Try
        'Obteniendo linea
        linea = e.RowIndex

        numOcupantesFaltan = Me.DataGridView2.RowCount
        'Controlo que al actualizar el numero de ocupantes, este no sea menor que el numero de filas del dg2
        If numOcupantesFaltan > Me.DataGridView1.Rows(e.RowIndex).Cells(4).Value() Then
            MessageBox.Show("ERROR: Número de ocupantes registrados mayor al indicado en este campo")
            Me.DataGridView1.Rows(e.RowIndex).Cells(4).Value() = numOcupantesHay
        End If

        'En caso de que cambien el numero de ocupantes doy la opcion de añadir nuevos
        If numOcupantesHay > numOcupantesFaltan Then
            Me.añadeOcupantes.Show()
            Me.añadeOcupantes.Enabled = True
            realizaConsultaOcupantes()
        Else
            Me.añadeOcupantes.Hide()
        End If

    Catch ex As Exception
    End Try
End Sub
```

## 3- Variable “TipoColision”

### Problema

La variable TipoColisión se rellena mediante una lista desplegable que ofrece una serie de opciones, entre las que esta “Vuelco sin colisión”. Esta opción no puede estar seleccionada en el accidente que sufre un vehículo que hemos almacenado en la

tabla "ChocaCon", tabla específica para accidentes consistentes en una colisión entre vehículos.

## Solución

Si pretendemos introducir una colisión entre vehículos en la tabla "ChocaCon" antes de aceptar esta inserción el sistema comprueba que ambos vehículos sufrieron el accidente en la misma fecha y a la misma hora, así como de qué tipo fue el accidente. Si la fecha o la hora son diferentes, o si la variable "TipoColision" es "Vuelco sin colisión" en alguno de ellos, el sistema indica al usuario mediante un pantallazo la imposibilidad de realizar esa inserción.

```
Private Sub guarda_colision_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles guarda_colision.Click
    Dim numero_selecciones As Integer = 0
    Dim matricula1 As String = "a", matricula2 As String = "b",
    fecha1 As String = "c", fecha2 As String = "d", hora1 As String = "e", hora2 As String = "f"

    Try
        For index As Integer = 0 To DataGridView1.RowCount - 1
            If (Me.DataGridView1.Rows(index).Cells(0).Value = True) Then
                If
                    (Me.DataGridView1.Rows(index).Cells(5).Value.ToString.Equals("Vuelco sin colisión") = False) Then
                        numero_selecciones = numero_selecciones + 1
                        If numero_selecciones = 1 Then
                            matricula1 = Me.DataGridView1.Rows(index).Cells(1).Value
                            fecha1 = Me.DataGridView1.Rows(index).Cells(2).Value
                            hora1 = Me.DataGridView1.Rows(index).Cells(3).Value
                        ElseIf numero_selecciones = 2 Then
                            matricula2 = Me.DataGridView1.Rows(index).Cells(1).Value
                            fecha2 = Me.DataGridView1.Rows(index).Cells(2).Value
                            hora2 = Me.DataGridView1.Rows(index).Cells(3).Value
                        Else
                            MessageBox.Show("ERROR: número de vehículos seleccionados superior a dos.")
                            Exit Sub
                        End If
                    Else
                        MessageBox.Show("ERROR: El vehículo " & Me.DataGridView1.Rows(index).Cells(1).Value & " sufrió un Vuelco sin Colisión.")
                        Exit Sub
                    End If
                End If
            End If
        Next
```

```

If numero_selecciones = 2 Then
    If fecha1.Equals(fecha2) = False Then
        MessageBox.Show("ERROR: colisión imposible, los
        vehículos resultaron accidentados en días
        diferentes.")
    Else
        If hora1.Equals(hora2) = False Then
            MessageBox.Show("ERROR: colisión imposible, los
            vehículos resultaron accidentados en horas
            diferentes.")
        Else
            Dim choca As InsertandoEnChocaCon = New
            InsertandoEnChocaCon
            choca.InsertarEnChocaCon(matricula1, fecha1,
            hora1, matricula2)
            choca.InsertarEnChocaCon(matricula2, fecha1,
            hora1, matricula1)
            Me.DataGridView1.Columns(0).Visible = False
            Me.guarda_colision.Hide()
            Me.añadir_colision.Show()
        End If
    End If
ElseIf numero_selecciones < 2 Then
    MessageBox.Show("ERROR: número de vehículos
    seleccionados inferior a dos.")
End If
Catch ex As Exception
End Try
End Sub

```

Una vez hayamos insertado una colisión en “ChocaCon” tenemos la opción de volver a visualizar los datos del accidente y, por lo tanto, podemos actualizar dichos datos. Como he dicho con anterioridad debo controlar que la variable “TipoColision” no se cambie por “Vuelco sin colisión” ya que sería ambiguo. Para ello genero una serie de funciones:

ANTES de actualizar:

```

Private Sub DataGridView1_CellBeginEdit(ByVal sender As Object, ByVal
e As System.Windows.Forms.DataGridViewCellCancelEventArgs) Handles
DataGridView1.CellBeginEdit
    Dim linea As Integer

    linea = e.RowIndex
    ColisionAntesCambio =
    Me.DataGridView1.Rows(linea).Cells(5).Value()
End Sub

```

DESPUÉS de actualizar, paso los datos a Comprueba\_tipo\_colision:

```

'Comprueba_cambio_en_tipo_colision(matricula, Fecha, Hora)
If (Me.DataGridView1.Rows(e.RowIndex).Cells(5).Value.Equals("Vuelco
sin colisión")) Then
    Dim tieneColision As String = comprueba_si_tiene_colision(matricula,
    Fecha, Hora)

```

```

If (tieneColision = 1) Then
    MessageBox.Show("ERROR: Este accidente esta almacenado como
colisión entre vehículos, no puedes cambiar el tipo de colisión
a 'Vuelco sin colisión'.")
Me.DataGridView1.Rows(e.RowIndex).Cells(5).Value =
ColisionAntesCambio
End If
End If

```

Comparativas para ver si el accidente esta en “ChocaCon”, para ello llama a consulta\_para\_colision():

```

Public Function comprueba_si_tiene_colision(ByVal mat As String, ByVal
fec As String, ByVal ho As String) As Integer
    Dim myDt As DataTable
    Dim siOno As Integer = 0

    myDt = consulta_para_colision()
    If (myDt.Rows.Count <> 0) Then
        Dim drow As DataRow
        For Each drow In myDt.Rows
            If (mat.Equals(drow("matricula"))) Then
                Dim FechaDate As DateTime = CType(drow("fecha"),
DateTime)
                Dim enDate As String = Format(FechaDate, "yyyy-MM-
dd")
                If (fec.Equals(enDate)) Then
                    If (ho.Equals(drow("hora"))) Then
                        siOno = 1
                    End If
                End If
            End If
        Next
    End If
    Return siOno
End Function

```

Consulta sobre “ChocaCon” para ver su contenido:

```

Public Function consulta_para_colision() As DataTable
    Dim DBCon As MySqlConnection
    Dim dt As DataTable

    'Abriendo conexion con la base de datos
    Dim myConString As MySqlConnectionString = New
MySQLConnectionString("localhost", "DatosAccidentes", "root",
"", 3309)

    DBCon = New MySqlConnection(myConString.AsString)

    Try
        DBCon.Open() 'Abrimos la conexión
        Dim atributos(2) As String
        atributos(0) = "matricula"
        atributos(1) = "fecha"
        atributos(2) = "Hora"

        Dim t(0) As String

```

```

t(0) = "ChocaCon"

'Preparando la consulta en la tabla chocacon
dt = New MySQLSelectCommand(DBCon, atributos, t, Nothing,
Nothing, Nothing).Table

Catch ex As Exception
End Try
Return dt
End Function

```

De esta forma compruebo si la actualización se puede realizar o no. En caso de no poderse se le informará al usuario mediante un pantallazo y se devolverá el valor de la variable al anterior a la actualización.

## **Capítulo 5**

# **INSTALACIÓN Y CONFIGURACIÓN**

## 5.1 - Instalación y configuración

El objetivo de este apartado es dejar reflejado los pasos básicos para realizar la instalación de todas las aplicaciones y su configuración necesaria. Además estos pasos ayudarán a los futuros usuarios a realizar correctamente los pasos previos a la utilización de los programas. Vamos a ir viendo por pasos la instalación de cada uno de los componentes de este proyecto.

### MySQL Server 6.0

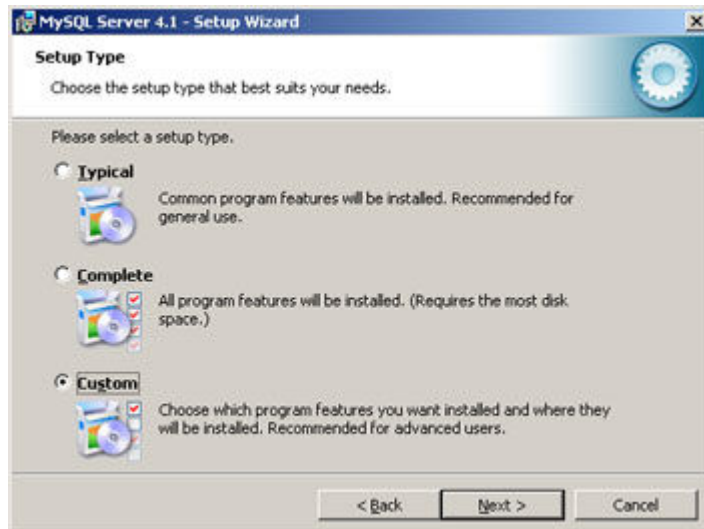
Es la única parte cuya instalación y configuración entraña cierta dificultad. Además su grado de importancia es crítico ya que es la base sobre la que actúan el resto de aplicaciones.

Este manual muestra paso a paso cómo instalar MySQL Server (Base de Datos SQL gratuita y muy difundida por Internet):

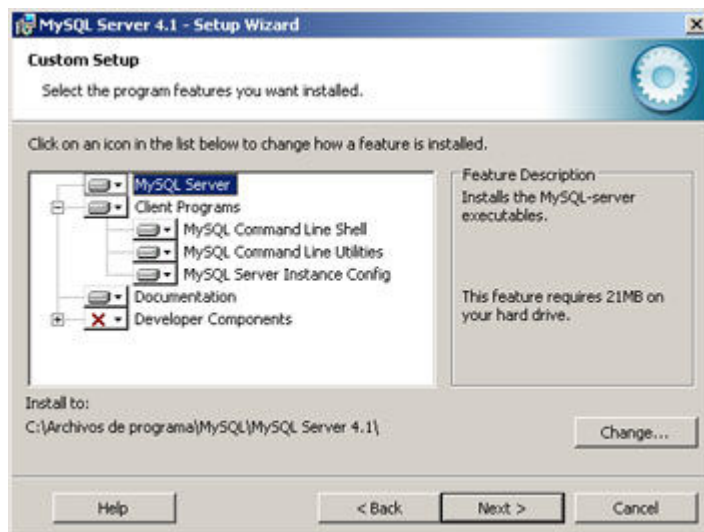
En primer lugar ejecutaremos el programa de instalación y seguiremos las instrucciones que nos muestra el asistente de instalación.



Pulsaremos en "Next" y marcaremos "Custom".

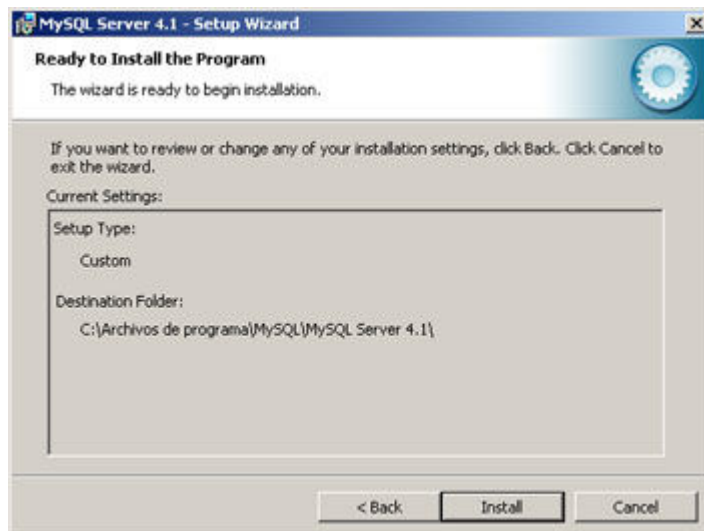


Seleccionamos las utilidades a instalar, por defecto se instalará todo salvo las herramientas para desarrolladores, innecesarias en este caso.



Pulsamos en "Next" y a continuación en "Install".



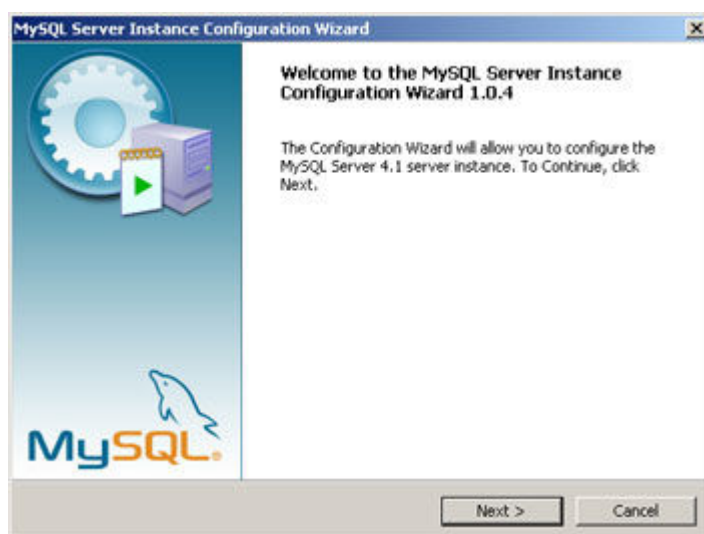


En la siguiente ventana tenemos dos opciones seleccionadas por defecto las cuales nos invitan a:

- 1- Registrarnos en MySQL.com o si ya estamos registrados introducir email de registro y contraseña. También podemos cancelar el registro, sabiendo que no es necesario para utilizar la aplicación. Pero en caso de que queramos hacerlo marcaremos en "Create a new free MySQL.com account" e iremos rellenando los datos que nos pide.
- 2- Configurar MySQL, opción que deberemos seleccionar. Para ello dejaremos marcada la opción "Configure the MySQL Server now".

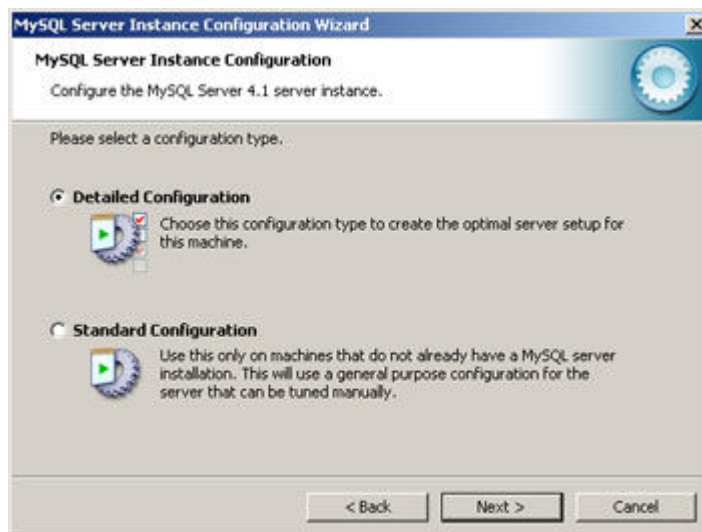
Una vez elegidas las opciones pulsaremos en "Finish".

Ahora nos aparecerá un asistente para la configuración "MySQL Server Instance Configuration Wizard" y pulsaremos en "Next":



*Nota: En caso de que no se inicie el asistente, desde el escritorio acudiremos al menú inicio, seleccionaremos programas y entre ellos buscaremos MySQL. Una vez localizado clicaremos en "MySQL Server Instance Configuration Wizard".*

Marcaremos la opción "Detailed Configuration" y pulsaremos en "Next", de esta forma podremos configurar más opciones de MySQL utilizando el asistente. Si marcásemos "Standard Configuration" el asistente nos pediría menos información pero habría que configurar algunas opciones manualmente:



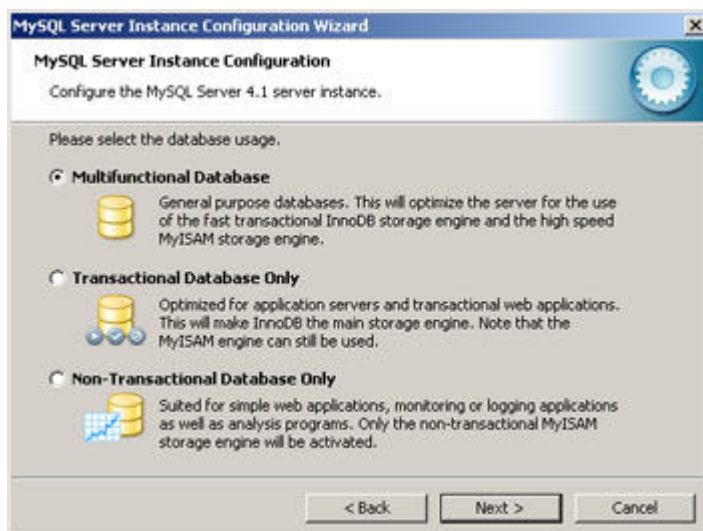
Dependiendo del uso que queramos dar al equipo en el que se instala marcaremos una de las tres opciones:

- **Developer Machine:** Opción predeterminada que seleccionaremos en caso de que el equipo donde hemos instalado MySQL Server vaya a ser utilizado también para otras aplicaciones. MySQL Server utilizará la memoria mínima necesaria.
- **Server Machine:** si vamos a utilizar el equipo para algunas aplicaciones (no demasiadas), a parte de MySQL. Con esta opción la aplicación utilizará un nivel medio de memoria, permitiendo la ejecución de otros programas.
- **Dedicated MySQL Server Machine:** marcaremos esta opción sólo si queremos utilizar el equipo como un servidor dedicado exclusivamente a MySQL. Con esta opción MySQL Server utilizará el máximo de memoria disponible.

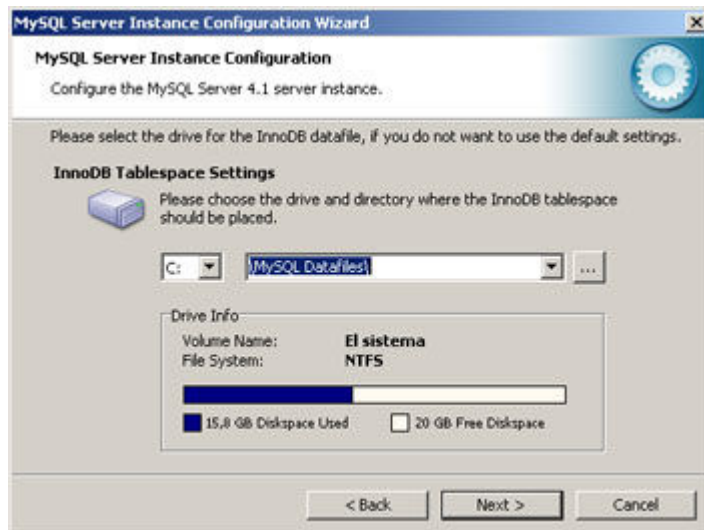
En nuestro caso marcaremos "Developer Machine" (consume el mínimo de memoria necesaria para su funcionamiento), este tipo de configuración de la instancia de MySQL no es recomendable si la base de datos va a soportar múltiples conexiones concurrentes con un volumen importante de información. Aunque puesto que nosotros la utilizaremos para una aplicación local será suficiente:



Dependiendo del uso que queramos dar a la Base de Datos marcaremos una de las tres opciones siguientes, nosotros marcaremos "Multifunctional Database" ya que no deseamos utilizar MySQL como base de datos para transacciones de otra Base de Datos MySQL, que es lo que te permiten el resto de opciones.



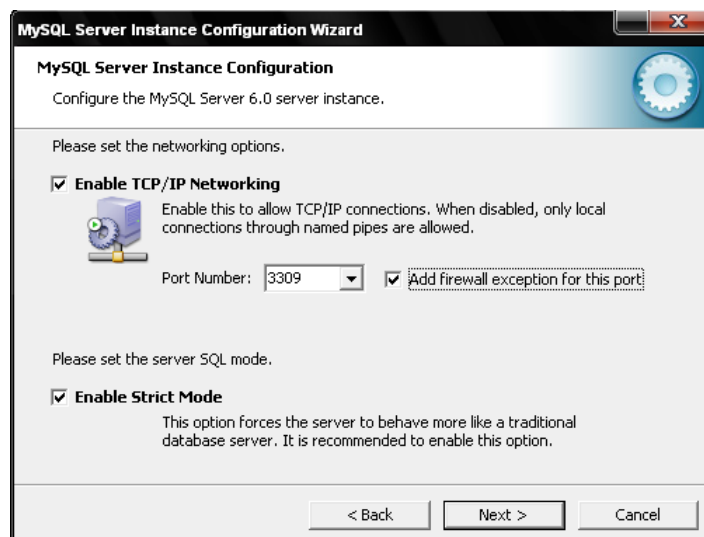
Seleccionaremos la unidad y la carpeta donde queramos guardar los ficheros de datos (Tablespace) de la Base de Datos. Para cambiar la opción por defecto ("C:\Archivos de programa\MySQL") deberemos clicar en "Modify".



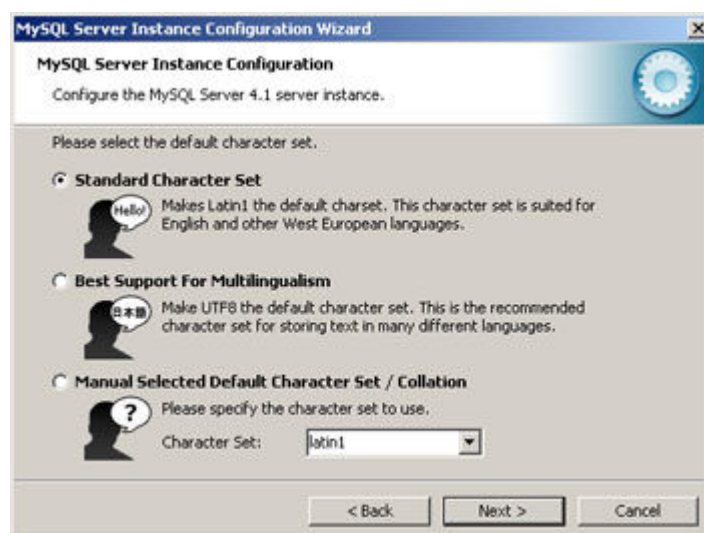
Seleccionaremos ahora el número aproximado de conexiones concurrentes (varios clientes conectados a la vez) que tendrá nuestro servidor de MySQL). La primera opción asume unas 20, la segunda unas 500 y la tercera permite especificarlas manualmente. Este parámetro es aproximado no tiene por qué ser exacto. En nuestro caso como el uso de la aplicación es local seleccionaremos la primera opción.



Dejaremos marcada la opción "Enable TCP/IP Networking" si queremos que los clientes se puedan conectar mediante TCP/IP al equipo servidor de MySQL. **Cambiaremos el puerto al 3309**, y seleccionaremos la opción de "Add firewall exception for this port" que añade una excepción al puerto indicado, abriéndolo para que la aplicación pueda utilizarlo.



Seleccionaremos el juego de caracteres que queramos utilizar, por defecto está marcado "Latin1" válido para Inglaterra y Europa:



El siguiente paso es importante pues nos pide que especifiquemos el tipo de arranque de MySQL Server. Si seleccionamos la primera opción ("Install As Windows Service") el programa de instalación nos creará un Servicio que será el encargado

de ejecutar MySQL Server, también nos permite especificar el nombre del servicio y si queremos que arranque automáticamente al iniciar el sistema ("Launch the MySQL Server automatically"). La segunda opción "Include Bin Directory in Windows PATH" añadirá las variables de entorno necesarias para la ejecución de los ficheros necesarios para iniciar MySQL.

La opción "Install As Windows Service", que es la que nos interesa, viene seleccionada por defecto.

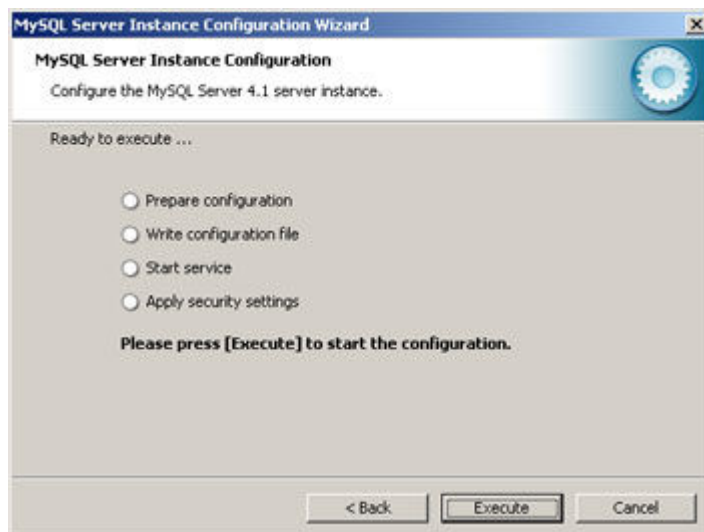


En nuestro caso no necesitamos ninguna contraseña, ya que utilizaremos MySQL de forma local, por lo que deseleccionaremos la primera opción. En caso de necesitarla introduciremos la contraseña para el usuario administrador (root) y marcaremos la opción "Enable root access from remote machines" si queremos que se pueda acceder como administrador desde otros equipos.





Por último pulsaremos en "Execute" para finalizar la configuración de MySQL:



Si no hay problemas mostrará esta ventana indicando que el proceso de instalación y configuración de MySQL Server ha terminado y se ha instalado e iniciado el Servicio que ejecutará MySQL:



## **Programa administrador**

Aplicación desarrollada para la gestión de la base de datos y de su contenido. Su instalación y configuración esta totalmente realizada. Lo único que deben tener en cuenta los usuarios es que la aplicación se debe ejecutar en la misma carpeta en la que están las librerías de MySQL, por lo que si el usuario desea tener el lanzador del programa en otra ubicación debe crea un acceso directo.

Las librerías de MySQL son:

- libmySQL-4.0.dll
- MySQLDriverCS.dll
- libmySQL.dll

## **Programa principal de gestión de accidentes**

Aplicación principal de mi proyecto. Desarrollada para la inserción y consulta de todos los datos referentes a accidentes de tráfico de forma sencilla y segura. Con respecto a su instalación y configuración ocurre lo mismo que el programa administrador: se debe ejecutar en la misma carpeta en la que tenemos las librerías de MySQL.



## **Capítulo 6**

# **CONCLUSIONES Y BIBLIOGRAFÍA**

## 6.1 - Conclusiones

Tras un largo periodo desarrollando todo lo explicado en estas páginas, diré que he aprendido bastante del trabajo que supone el desarrollo de un proyecto desde sus fases iniciales en las que asenté las bases del proyecto mediante entrevistas con el cliente, al arduo trabajo que suponen las pruebas de la aplicación con el objetivo de encontrar errores.

Cada una de las fases tiene su propia dificultad y en todas ellas he tenido que pelear para sacarlas adelante:

- El periodo de análisis de requisitos marca las bases sobre las que luego debes desarrollar todo, por lo que si deseas hacer un buen trabajo debes dejar muy claro lo que deseas desarrollar y como vas a hacerlo. Si no realizas un buen análisis, a posteriori va a suponer muchas horas de trabajo perdidas y muchas líneas de código desechadas.
- El proceso de implementación, para mi gusto ha sido el más apasionante. Mi forma de verlo se asemeja a un gran puzzle, con piezas infinitas que debes unir para obtener una imagen. Las ideas y las dudas van fluyendo, precisando de un gran estudio de las posibilidades del lenguaje. Cada punto es una nueva lucha que necesita de mucha imaginación para sacarlo adelante.
- Las pruebas del producto “terminado” me llegaron a desesperar. Por más que corregía errores siempre quedaban pequeños flecos, siempre se me ocurría algo nuevo que desarrollar o que mejorar.

Sin más dilaciones y para finalizar esta memoria diré que agradezco la oportunidad de realizar un trabajo que va a tener una utilidad real, el apoyo de mi tutora Eurne Barrenechea y la gran ayuda de Diego Reyero (al que llamaré mi cliente) en todos los temas médicos que he tenido que introducir en los programas.

## 6.2 – Bibliografía

### Artículos:

Diego Reyero Díez, MD,\* Tomás Belzunegui Otano, MD,† Begoña Bermejo Fraile, MD,‡ Clint Jean Louis, MD,† Jose Roldán Ramírez, MD,§ and Alfredo Echarri Sucunza, MD\*.

USE OF A STRUCTURAL DEFORMITY INDEX AS A PREDICTOR OF SEVERITY AMONG TRAUMA VICTIMS IN MOTOR VEHICLE CRASHES (pp, 1-10).

### Tutoriales:

Miguel Pagola Barrio.

TEORÍA Y PRACTICAS DE GESTIÓN DE SISTEMAS (2008/2009).  
Capítulo principal: ADO .NET

Luis Miguel Blanco.

PROGRAMACIÓN EN VISUAL BASIC .NET.

### Páginas Web:

<http://msdn.microsoft.com/es-es/library/> (**librerías**)

[http://grupos.emagister.com/foro\\_visual\\_studio/7227](http://grupos.emagister.com/foro_visual_studio/7227) (**Foro de consultas varias**)

<http://www.google.es/> (**Buscador principal**)

# DISEÑO DE UNA APLICACIÓN PARA EL ESTUDIO DE LA BIOGINÉTICA EN LOS ACCIDENTES DE TRÁFICO

# Introducción

- Los accidentes de tráfico en el año 2009 provocaron 1.897 muertos, en un número de accidentes que cuadriplica la cifra anterior.
- Recursos materiales y humanos para combatirlos o atenderlos:
  - Cuerpos de seguridad del estado
  - Bomberos
  - Voluntariado (Bomberos voluntarios, DYA, Protección Civil...)
  - Servicios médicos de emergencias

# Introducción II

- Criterio de actuación:

Los primeros en llegar al accidente (normalmente policía) valoran a los accidentados y avisan a los servicios médicos. Según dicha valoración se selecciona el equipamiento médico.

- Problema:

La valoración de los accidentados por parte de personal no titulado en medicina.

- Origen del proyecto:

Necesidad de un método de valoración de la gravedad de los accidentados basado en un conjunto de factores apreciables a simple vista (Tesis del IDE). Para su desarrollo es necesario el estudio de infinidad de datos.

# Introducción / estado del arte

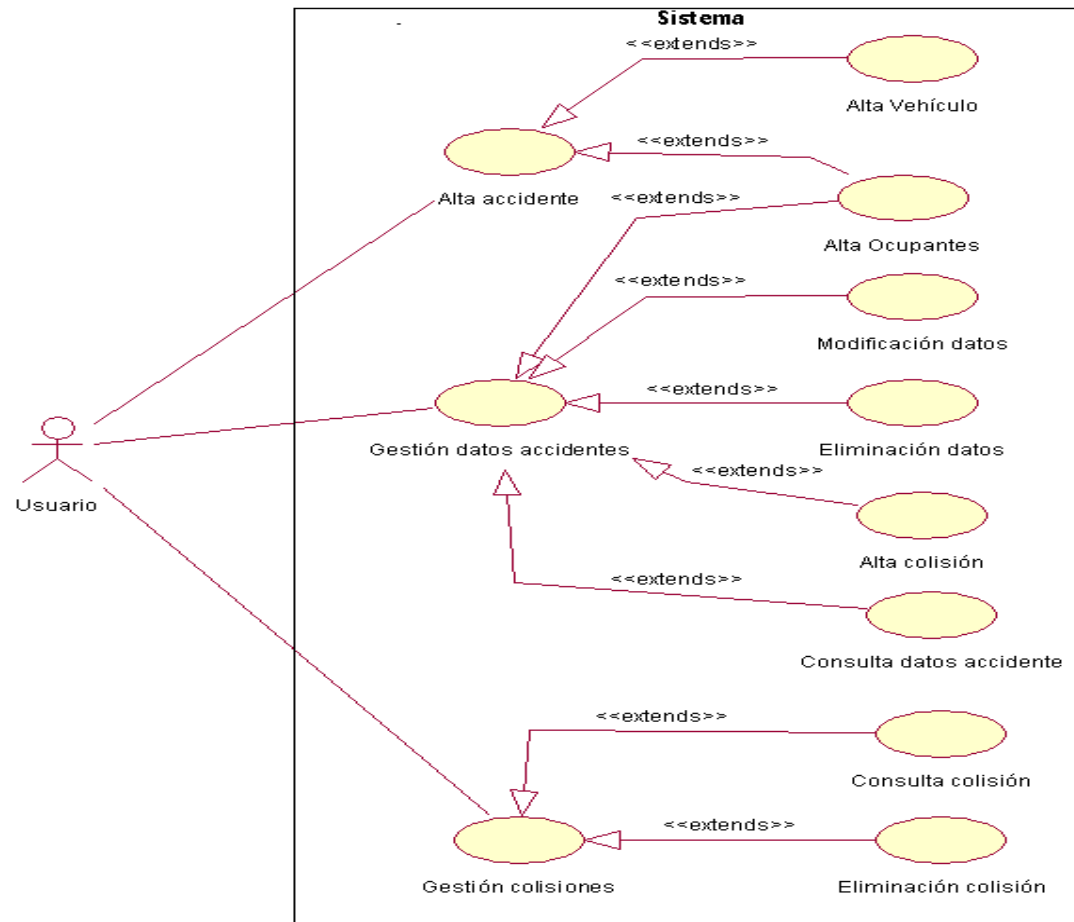
- Actualmente existen muchas herramientas para el almacenamiento y tratamiento de datos (Excel, Spss,...)
- Problemas:
  - Gran redundancia de los datos
  - Ralentización del trabajo de tratamiento y estudio de la información
  - Falta de especialización provoca mayor dificultad de uso.

# Introducción / Objetivo del Proyecto

- Una aplicación para el almacenamiento y consulta específico para accidentes de tráfico y sus ocupantes.
- Control de datos y una ordenación supervisada y evaluada por el desarrollador de la tesis.
- Interfaz gráfica, sencilla y robusta.
- Facilitar el acceso a datos concretos, facilitando su estudio

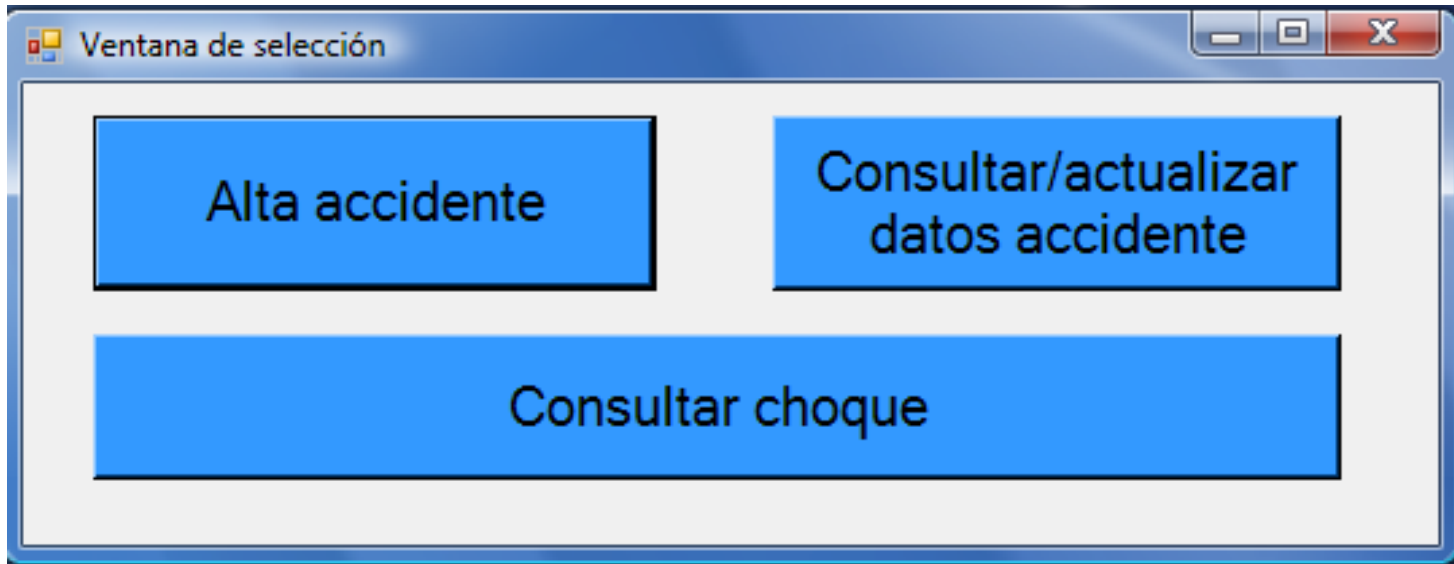


# Casos de uso / Programa principal

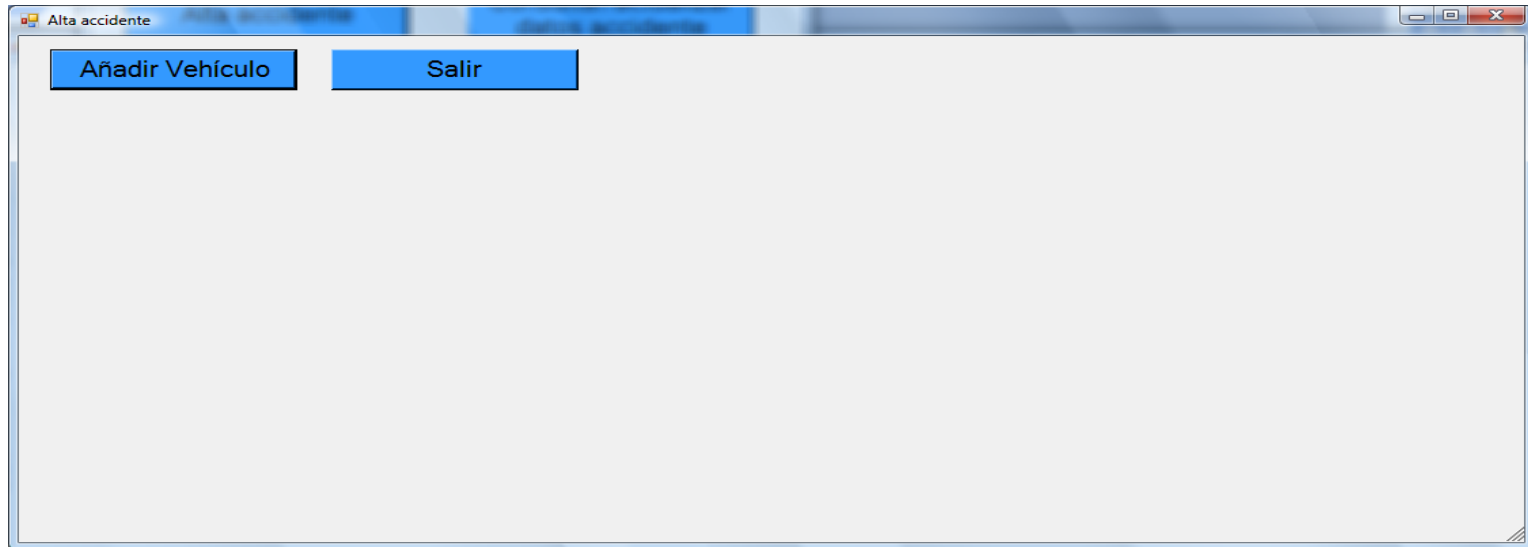


# Casos de uso / Programa principal (II)

## Ventana de selección



# Casos de uso / Alta accidente



- Compuesto por:
  - Alta vehículo
  - Alta ocupantes

# Casos de uso / Alta accidente (II)

- Alta vehículo

Datos Vehículo

*Los campos marcados con \* son obligatorios*

\* Fecha del accidente:  Año-Mes-Dia(XXXX-XX-XX)

\* Hora:  Horas:Minutos(XX:XX)

\* Matrícula:

\* Número de ocupantes:

Tipo de colisión:

Tipo de Vehículo:

Año de matriculación del vehículo:

Número de puntos EuronCAP:

¿La velocidad a la que circulaba estaba por encima de lo permitido?

☐ Si ☒ No ☐ Vel. Desconocida

¿El vehículo tenía airbag?

☐ Si ☒ No

# Casos de uso / Alta accidente (III)

Alta accidente

	Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AñoMatriculacion
▶	1250CWC	2010-06-22	12:00	2	Alcance ▼	Si ▼	Turismo(pequeño) ▼	80	Si ▼	4	Entre 5 y 10 años ▼

Añadir Ocupante

Cancelar Guardar

# Casos de uso / Alta accidente (IV)

- Alta ocupantes

DatosOcupantesModificado

*Los campos marcados con "\*" son obligatorios*

\* Edad:

\* IDE:

Glasgowr samu:

Glasgowr urg:

Glasgowr uci:

RTS samu:

RTS urg:

RTS uci:

ISS urg:

ISS uci:

Lesión principal:

\* ¿Qué tipo de atención recibió?

☐ Forense   ☐ UCI   ☐ NoETNA

tce: ☐ Si   ☒ No

tfacial: ☐ Si   ☒ No

traquis: ☐ Si   ☒ No

ttorax: ☐ Si   ☒ No

tabd: ☐ Si   ☒ No

tee: ☐ Si   ☒ No

tbland: ☐ Si   ☒ No

Cancelar   Aceptar

# Casos de uso / Alta accidente (V)

Alta accidente

	Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AñoMatriculacion
▶	1250CWC	2010-06-21	12:00	2	Alcance ▼	Si ▼	Turismo(pequeño) ▼	80	Si ▼	4	Entre 5 y 10 años ▼

**Añadir Ocupante**

	Edad	IDE	Glasgowssamu	Glasgowwurg	Glasgowwuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	LesionPrincipal	TipoAtencion	Tce	Tfe
▶	23	2	2	1			1	1			laceraciones en las extremidades inferiores	NoEtna ▼	No ▼	No

Cancelar Guardar

# Casos de uso / Gestión datos accidentes

Muestr/actualiza accidentea

Matricula:  Fecha:  Hora:

**Consultar** **Salir**

*NOTA: Para eliminar una linea haz click con el boton derecho sobre ella*

- Compuesto por:
  - Consulta datos accidente
  - Modificación datos
  - Eliminación datos
  - Alta ocupantes
  - Alta colisión



# Casos de uso / Gestión datos accidentes (II)

- Consulta datos accidente
- Modificación datos

Muestr/actualiza accidentea

Matricula:  Fecha:  Hora:

*NOTA: Para eliminar una linea haz click con el boton derecho sobre ella*

Matricula	umOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
1250CWC		Alcance	Si	Turismo(pequeño)	80	Si	4	Entre 5 y 10 años
0001BBC								
1234ABC								
2211AAB								
3344BBC								
4444GFD								

Edad	IDE	GlasgowSamu	Glasgowurg	Glasgowuci	RTSamu	RTSurg	RTSuci	ISSurg	ISSuci	LesionPrincipal	TipoAtencion	Tce	Tfa
23	2	2	1			1	1			laceraciones en las extremidades inferiores	NoEtna	No	No

# Casos de uso / Gestión datos accidentes (III)

- Eliminación datos

Muestr/actualiza accidentea

Matricula:  Fecha:  Hora:

*NOTA: Para eliminar una linea haz click con el boton derecho sobre ella*

	Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
▶	1250CWC	21/06/2010	12:00	2	Alcance	Si	Turismo(pequeño)	80	Si	4	Entre 5 y 10 años

Eliminar Linea	Glasgowssamu	Glasgowwurg	Glasgowwuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	LesionPrincipal	TipoAtencion	Tce	Ttra
	2	1			1	1			laceraciones en las extremidades inferiores	NoEtra	No	No

# Casos de uso / Gestión datos accidentes (IV)

## ● Alta ocupantes

*Los campos marcados con "\*" son obligatorios*

\* Edad:

\* IDE:

Glasgowr samu:

Glasgowr urg:

Glasgowr uci:

RTS samu:

RTS urg:

RTS uci:

ISS urg:

ISS uci:

Lesión principal:

\* ¿Qué tipo de atención recibió?

☐ Forense ☐ UCI ☐ NoETNA

tce: ☐ Si ☒ No

tfacial: ☐ Si ☒ No

traquis: ☐ Si ☒ No

ttorax: ☐ Si ☒ No

tabd: ☐ Si ☒ No

tee: ☐ Si ☒ No

tbland: ☐ Si ☒ No

Cancelar Aceptar

# Casos de uso / Gestión datos accidentes (V)

- Alta colisión

Muestr/actualiza accidentea

Matricula:  Fecha: 2010-06-04 Hora:

*NOTA: Para eliminar una línea haz click con el boton derecho sobre ella*

	Colisión	Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0001BBC	04/06/2010	12:00	2	Alcance	Desconocida	Turismo(pequeño)	0	No	2	Entre 5 y 10 años
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1234ABC	04/06/2010	12:00	2	Lateral	Desconocida	Todoterreno	0	No	2	Menos de 5 años
<input type="checkbox"/>	<input type="checkbox"/>	4444GFD	04/06/2010	12:00	1	Lateral	No	Turismo(pequeño)	80	Si	2	Más de 10 años

	Edad	IDE	Glasgowssamu	Glasgowurg	Glasgowuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	LesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
▶	21	2	2	2			2	2			heridas leves en la cara	NoEtna	Si	Si	No
	21	2	2				1		1	1		NoEtna	No	No	No

# Casos de uso / Gestión colisiones

MuestraChoques

Matricula:  Fecha:  Hora:

*NOTA: Para eliminar una línea haz click con el boton derecho sobre ella*

Resto de vehículos implicados:

- Compuesto por:
  - Consulta colisión
  - Eliminación colisión

# Casos de uso / Gestión colisiones (II)

- Consulta colisión

MuestraChoques

Matricula:  Fecha:  Hora:

*NOTA: Para eliminar una linea haz click con el boton derecho sobre ella*

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
0001BBC	04/06/2010 0:00:00	12:00	2	Vuelco sin colisión	Desconocida	Turismo grande	0	No	2	Entre 5 y 10 años

Edad	IDE	Glasgowsamu	Glasgowurg	Glasgowuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	lesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
23	3	3	3				3	3	3		NoEtna	No	Si	No

Resto de vehículos implicados:

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
1234ABC	04/06/2010 0:00:00	12:00	2	Lateral	Desconocida	Todoterreno	0	No	2	Menos de 5 años

Edad	IDE	Glasgowsamu	Glasgowurg	Glasgowuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	lesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
21	2	2	2			2	2			heridas leves en la cara	NoEtna	Si	Si	No
21	2	2				1		1	1		NoEtna	No	No	No

# Casos de uso / Gestión colisiones (III)

- Eliminación colisión

MuestraChoques

Matricula: 0001BBC Fecha: 2010-06-04 Hora: 12:00 **Consultar** **Salir**

*NOTA: Para eliminar una linea haz click con el boton derecho sobre ella*

Eliminar Linea	Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
▶	0001BBC	04/06/2010 0:00:00	12:00	2	Vuelco sin colisión	Desconocida	Turismo grande	0	No	2	Entre 5 y 10 años

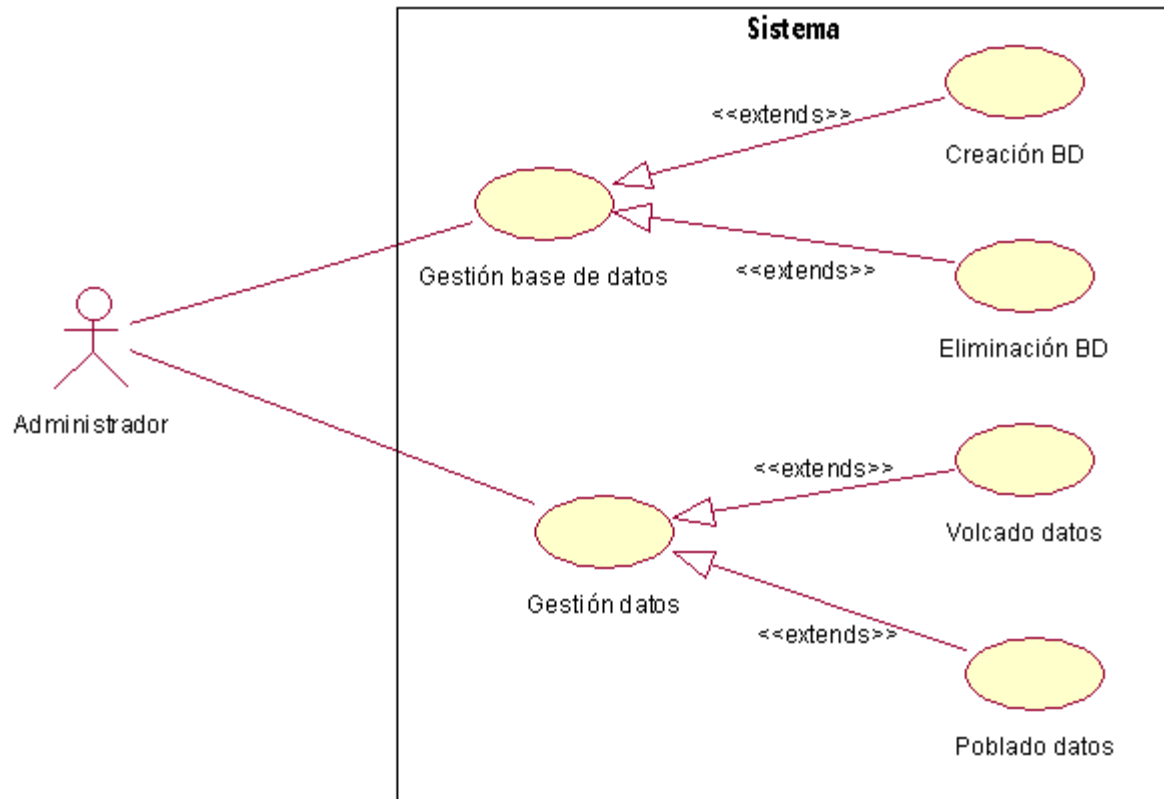
Edad	IDE	GlasgowSamu	Glasgowurg	Glasgowuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	lesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
▶ 23	3	3	3				3	3	3		NoEtna	No	Si	No

Resto de vehículos implicados:

Matricula	Fecha	Hora	NumOcupantes	TipoColision	VelocidadExcesiva	TipoVehiculo	VelocidadEstimada	Airbag	EuronCAP	AnioMatriculacion
▶ 1234ABC	04/06/2010 0:00:00	12:00	2	Lateral	Desconocida	Todoterreno	0	No	2	Menos de 5 años

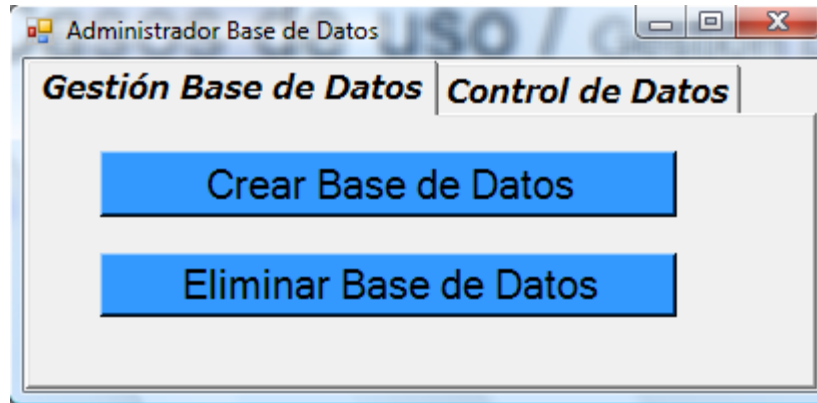
Edad	IDE	GlasgowSamu	Glasgowurg	Glasgowuci	RTSsamu	RTSurg	RTSuci	ISSurg	ISSuci	lesionPrincipal	TipoAtencion	Tce	Tfacial	Traquis
▶ 21	2	2	2			2	2			heridas leves en la cara	NoEtna	Si	Si	No
21	2	2				1		1	1		NoEtna	No	No	No

# Casos de uso / Administrador



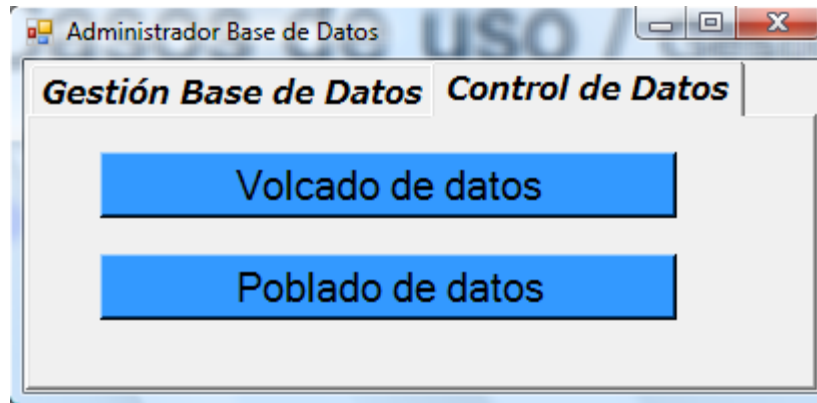


# Casos de uso / Gestión base de datos



- Compuesto por:
  - Creación BD
  - Eliminación BD

# Casos de uso / Gestión datos



- Compuesto por:
  - Volcado datos
  - Poblado dato

# Implementación / Herramientas

- Visual Studio .NET
- MySQL Server 6.0
- Librerías:
  - libmySQL-4.0.dll
  - MySQLDriverCS.dll
  - libmySQL.dll



# Ruegos y Preguntas